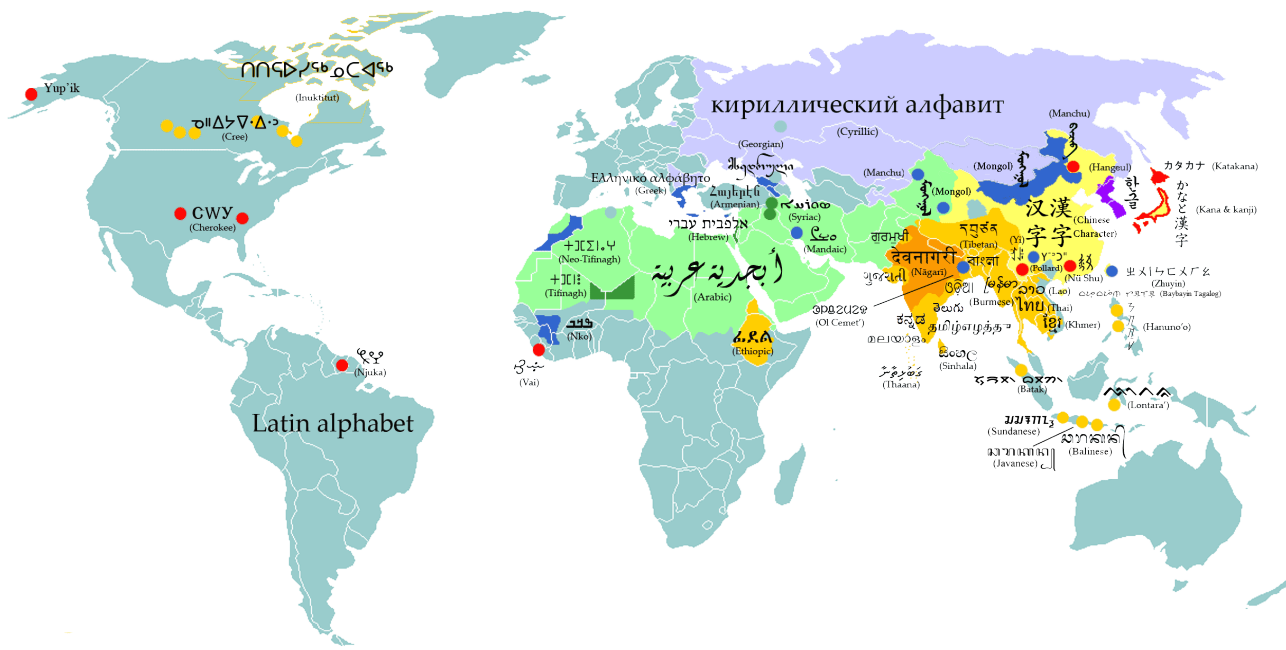
On the Effective Deployment of Current Machine Translation Technology

Jesús González Rubio



UNIVERSITAT POLITÈCNICA DE VALÈNCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

On the Effective Deployment of Current Machine Translation Technology

By
Jesús González Rubio

written under the direction of
Dr. Daniel Ortiz Martínez and Prof. Francisco Casacuberta

May 8, 2014

On the Effective Deployment of Current Machine Translation Technology

By

Jesús González Rubio

A thesis dissertation submitted to the
Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València
in partial fulfillment of the requirements
for the degree of
Doctor en Informática
written under the direction of
Dr. Daniel Ortiz-Martínez and Prof. Francisco Casacuberta
Valencia, Spain, May 8, 2014

Work supported by the European Union 7th Framework Program (FP7/2007-2013) under the CasMaCat project (grants agreement n^{o} 287576), and by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018) and the FPU scholarship AP2006-00691.

A mi madre por su paciencia infinita
y a mi padre que no pudo verla terminada

Copyright Note

I hereby declare that this dissertation describes my own original work developed under the direction of my two supervisors Dr. Daniel Ortiz-Martínez and Prof. Francisco Casacuberta. Further, I have acknowledged all sources used and have cited these in the reference section.

The work described in Chapter 3, however, is a result of a collaboration between me and José Ramón Navarro-Cerdán. The research interest of J. Ramón Navarro-Cerdán is focused on statistics. His part in the collaboration was to provide the background in statistical multivariate analysis required to develop the dimensionality reduction methods described in that chapter. For my part, I was in charge of applying this statistical background to the task of interest. This includes integrating the different dimensionality reduction methods into the data-flow of the system, implementing the heuristic feature selection methods described in the chapter, designing and performing the experimentation, and writing the publications derived from this research line.

Lastly, the different pieces of software that implement the developments described in this dissertation have been mostly implemented over previous libraries and translation toolboxes. I have acknowledged all these previous software by citing them in the reference section.

Agradecimientos

La elaboración de una tesis doctoral es un camino solitario que, paradójicamente, no podría recorrerse sin la colaboración de un gran grupo de personas. En estas líneas quisiera agradecer su apoyo a todos aquellos que de una forma u otra han contribuido a la consecución de esta tesis.

En primer lugar quisiera dar las gracias a todos mis compañeros del PRHLT que, tras todos estos años, se han convertido en una segunda familia para mi. Especialmente a mis directores: a Paco que me dio la oportunidad de dedicarme a la investigación por lo que siempre le estaré agradecido y a Dani que me ha guiado para superar los vericuetos del dia a dia de la investigación académica.

Digo PRHLT pero debería decir UPV ya que sin la gente del ITI y del DSIC estos años habrían sido mucho más arduos y, sin duda alguna, mucho menos excitantes. Las discusiones durante el café o las pausas tomando el sol en la terraza son momentos que siempre me acompañarán.

Por supuesto, no puedo dejar de dar las gracias a todos los amigos que he ido haciendo desde que vine a Valencia a estudiar Informática allá por el año 2000: a mis compañeros de carrera que me acompañaron en los primeros años, a los miembros de ITI-spam por su flujo inacabable de propuestas de procrastinación, a la pandi de hijos de puta por acogerme como uno más y ser unos excelentes compañeros de fiesta, y a Philipp, Sonia y Toni por su amistad incondicional todos estos años.

Y, hablando de amistad, no puedo olvidarme de aquellos a los que conocí mucho antes de venir a Valencia y que, por fortuna, aun puedo seguir llamando amigos. Gente, no podría haberlo hecho sin vosotros.

Finalmente, me gustaría dar las gracias a mi familia que siempre me apoyó. Especialmente a mis padres que lo dieron todo para que yo pudiese llegar a dónde estoy ahora.

A todos, gracias.

Valencia 2014

Abstract

Machine translation is a fundamental technology that is gaining more importance each day in our multilingual society. Companies and particulars are turning their attention to machine translation since it dramatically cuts down their expenses on translation and interpreting. However, the output of current machine translation systems is still far from the quality of translations generated by human experts. The overall goal of this thesis is to narrow down this quality gap by developing new methodologies and tools that improve the broader and more efficient deployment of machine translation technology.

We start by proposing a new technique to improve the quality of the translations generated by fully-automatic machine translation systems. The key insight of our approach is that different translation systems, implementing different approaches and technologies, can exhibit different strengths and limitations. Therefore, a proper combination of the outputs of such different systems has the potential to produce translations of improved quality. We present minimum Bayes' risk system combination, an automatic approach that detects the best parts of the candidate translations and combines them to generate a consensus translation that is optimal with respect to a particular performance metric. We thoroughly describe the formalization of our approach as a weighted ensemble of probability distributions and provide efficient algorithms to obtain the optimal consensus translation according to the widespread BLEU score. Empirical results show that the proposed approach is indeed able to generate statistically better translations than the provided candidates. Compared to other state-of-the-art systems combination methods, our approach reports similar performance not requiring any additional data but the candidate translations.

Then, we focus our attention on how to improve the utility of automatic translations for the end-user of the system. Since automatic translations are not perfect, a desirable feature of machine translation systems is the ability to predict at run-time the quality of the generated translations. Quality estimation is usually addressed as a regression problem where a quality score is predicted from a set of features that represents the translation. However,

although the concept of translation quality is intuitively clear, there is no consensus on which are the features that actually account for it. As a consequence, quality estimation systems for machine translation have to utilize a large number of weak features to predict translation quality. This involves several learning problems related to feature collinearity and ambiguity, and due to the "curse" of dimensionality. We address these challenges by adopting a two-step training methodology. First, a dimensionality reduction method computes, from the original features, the reduced set of features that better explains translation quality. Then, a prediction model is built from this reduced set to finally predict the quality score. We study various reduction methods previously used in the literature and propose two new ones based on statistical multivariate analysis techniques. More specifically, the proposed dimensionality reduction methods are based on partial least squares regression. The results of a thorough experimentation show that the quality estimation systems estimated following the proposed two-step methodology obtain better prediction accuracy that systems estimated using all the original features. Moreover, one of the proposed dimensionality reduction methods obtained the best prediction accuracy with only a fraction of the original features. This feature reduction ratio is important because it implies a dramatic reduction of the operating times of the quality estimation system.

An alternative use of current machine translation systems is to embed them within an interactive editing environment where the system and a human expert collaborate to generate error-free translations. This interactive machine translation approach have shown to reduce supervision effort of the user in comparison to the conventional decoupled post-edition approach. However, interactive machine translation considers the translation system as a passive agent in the interaction process. In other words, the system only suggests translations to the user, who then makes the necessary supervision decisions. As a result, the user is bound to exhaustively supervise every suggested translation. This passive approach ensures error-free translations but it also demands a large amount of supervision effort from the user.

Finally, we study different techniques to improve the productivity of current interactive machine translation systems. Specifically, we focus on the development of alternative approaches where the system becomes an active agent in the interaction process. We propose two different active approaches. On the one hand, we describe an active interaction approach where the system informs the user about the reliability of the suggested translations. The hope is that this information may help the user to locate translation errors thus improving the overall translation productivity. We propose different scores to measure

translation reliability at the word and sentence levels and study the influence of such information in the productivity of an interactive machine translation system. Empirical results show that the proposed active interaction protocol is able to achieve a large reduction in supervision effort while still generating translations of very high quality. On the other hand, we study an active learning framework for interactive machine translation. In this case, the system is not only able to inform the user of which suggested translations should be supervised, but it is also able to learn from the user-supervised translations to improve its future suggestions. We develop a value-of-information criterion to select which automatic translations undergo user supervision. However, given its high computational complexity, in practice we study different selection strategies that approximate this optimal criterion. Results of a large scale experimentation show that the proposed active learning framework is able to obtain better compromises between the quality of the generated translations and the human effort required to obtain them. Moreover, in comparison to a conventional interactive machine translation system, our proposal obtained translations of twice the quality with the same supervision effort.

Resumen

La traducción automática es una tecnología fundamental que cada día está ganando más importancia en nuestra sociedad plurilingüe. Compañías y particulares están volviendo su atención hacia la traducción automática ya que les permite reducir dramáticamente sus gastos en traducción e interpretación. Sin embargo, las traducciones generadas por los actuales sistemas de traducción automática está aún lejos de la calidad de las traducciones generadas por traductores expertos. El objetivo general de esta tesis es reducir este salto de calidad mediante el desarrollo de nuevas metodologías y herramientas que permitan un despliegue más amplio y eficiente de la tecnología actual en traducción automática.

Comenzamos proponiendo una nueva técnica para mejorar la calidad de las traducciones generadas por los sistemas de traducción automática. La idea clave de nuestra propuesta es que diferentes sistemas de traducción, que implementen diferentes enfoques o tecnologías, pueden mostrar diferentes puntos fuertes y limitaciones. Por lo tanto, una combinación adecuada de las salidas de diferentes sistemas puede producir traducciones de mayor calidad. Presentamos *minimum Bayes' risk system combination*, una propuesta que automáticamente detecta las mejores partes de las traducciones candidatas y las combina para generar una traducción consenso que es óptima respecto a una medida particular de calidad. Describimos en profundidad la formalización de nuestro enfoque como una suma ponderada de distribuciones de probabilidad y proporcionamos algoritmos eficientes para obtener la traducción consenso óptima de acuerdo con la conocida medida de evaluación BLEU. Los resultados empíricos muestran que el método propuesto es realmente capaz de generar traducciones estadísticamente mejores que las traducciones candidatas proporcionadas. En comparación con otros métodos de combinación de sistemas del estado del arte, nuestro método obtiene un rendimiento similar no requiriendo información adicional mas allá de las traducciones candidatas.

A continuación, centramos nuestra atención en como mejorar la utilidad de las traducciones automáticas para el usuario final del sistema. Dado que las traducciones automáticas no son perfectas, una característica deseable de

los sistemas de traducción automática es la habilidad de predecir en tiempo de ejecución la calidad de las traducciones generadas. La estimación de la calidad se aborda generalmente como un problema de regresión en el que se predice un valor de calidad a partir de un conjunto de características que representan la traducción. Sin embargo, aunque el concepto de calidad en la traducción es intuitivamente claro, no existe consenso sobre cuales son las características que realmente dan cuenta de él. Como consecuencia, los sistemas de estimación de la calidad para traducción tienen que utilizar un gran número de características débiles para predecir la calidad de las traducciones. Esto implica diversos problemas de aprendizaje relacionados con la colinearidad y ambigüedad de las características, y también debidos a la "maldición" de la dimensionalidad. Nosotros abordamos estos retos adoptando una metodología de entrenamiento en dos pasos. En primer lugar, un método de reducción de la dimensionalidad calcula a partir de las características originales el conjunto de características que mejor explica la calidad de las traducciones. A continuación, construimos un modelo de predicción a partir de este conjunto reducido de características para predecir finalmente el valor de calidad. Estudiamos varios métodos de reducción previamente utilizados en la literatura y proponemos dos nuevos métodos basados en técnicas estadísticas de análisis multivariante. Más específicamente, los métodos propuestos para reducir la dimensionalidad se basan en la regresión por mínimos cuadrados parciales (en inglés, *partial least squares regression*). Los resultados de una experimentación exhaustiva muestran que los sistemas de estimación de la calidad estimados siguiendo la metodología en dos pasos propuesta obtienen una predicción más precisa que los sistemas estimados utilizando todas las características originales. Lo que es más, uno de los métodos propuestos para reducir la dimensionalidad obtuvo las predicciones más precisas necesitando sólo una fracción de las características originales. Este ratio de reducción en el número de características es particularmente importante porque implica una reducción drástica en los tiempos de respuesta del sistema de estimación de la calidad.

Un uso alternativo de los sistemas de traducción automática actuales es incorporarlos a un entorno interactivo de edición en el que el sistema y un usuario experto colaboran para generar traducciones correctas. Esta traducción automática interactiva ha demostrado ser capaz de reducir el esfuerzo de supervisión del usuario en comparación con un sistema de post-edición desacoplado. Sin embargo, la traducción automática interactiva considera al sistema de traducción un agente pasivo en el proceso interactivo. En otras palabras, el sistema sólo sugiere traducciones al usuario quien entonces toma las decisiones de supervisión necesarias. Como resultado, el usuario está obligado a supervisar

exhaustivamente cada traducción sugerida. Esta metodología pasiva asegura la obtención de traducciones sin errores pero también exige un gran esfuerzo de supervisión por parte del usuario.

Finalmente, estudiamos diferentes técnicas para mejorar la productividad de los sistemas actuales de traducción automática interactiva. Específicamente, nos centramos en el desarrollo de metodologías alternativas en las que el sistema se convierte en un agente activo en el proceso interactivo. Proponemos dos metodologías activas diferentes. Por un lado, describimos una metodología activa de interacción en la que el sistema informa al usuario sobre la fiabilidad de las traducciones sugeridas. Nuestra intuición es que esta información puede ayudar al usuario a localizar los errores de traducción, mejorando por lo tanto la productividad del sistema. Proponemos diferentes valores para medir la fiabilidad de las traducciones tanto a nivel de palabra como a nivel de traducción completa y estudiamos la influencia que tiene dicha información en la productividad de un sistema de traducción automática interactiva. Los resultados empíricos muestran que el protocolo activo de interacción propuesto es capaz de lograr grandes reducciones en el esfuerzo de supervisión y, al mismo tiempo, generar traducciones de muy alta calidad. Por otro lado, estudiamos un marco de aprendizaje activo para la traducción automática interactiva. En este caso, el sistema no sólo es capaz de informar al usuario sobre que traducciones deberían ser supervisadas sino que también es capaz de aprender las traducciones supervisadas por el usuario de forma que mejora sus sugerencias futuras. Desarrollamos un criterio de valor-de-información para seleccionar las traducciones automáticas que deberían ser supervisadas por el usuario. Sin embargo, dada su alta complejidad computacional, en la practica estudiamos diferentes estrategias de selección que aproximan este criterio óptimo. Los resultados de una experimentación a gran escala muestran que el marco de aprendizaje activo propuesto es capaz de obtener mejores compromisos entre la calidad de las traducciones generadas y el esfuerzo de supervisión requerido para obtenerlas. Lo que es más, en comparación con un sistema de traducción automática interactiva convencional, nuestra propuesta obtiene traducciones del doble de calidad con el mismo esfuerzo de supervisión.

Resum

La traducció automàtica és una tecnologia fonamental que cada dia està guanyant més importància en la nostra societat plurilingüe. Companyies i particulars estan tornant la seva atenció cap a la traducció automàtica ja que redueix dramàticament les seves despeses en traducció i interpretació. No obstant això, la sortida dels actuals sistemes de traducció automàtica està encara lluny de la qualitat de les traduccions generades per traductors experts. L'objectiu general d'aquesta tesi és reduir aquesta diferència de qualitat mitjançant el desenvolupament de noves metodologies i eines que permeten un desplegament més ampli i eficient de la tecnologia en traducció automàtica.

Comencem proposant una nova tècnica per millorar la qualitat de les traduccions generades pels sistemes de traducció automàtica. La idea clau de la nostra proposta és que diferents sistemes de traducció, que implementen diferents enfocaments o tecnologies, poden mostrar diferents punts forts i limitacions. Per tant, una combinació adequada de les sortides de diferents sistemes pot produir traduccions de major qualitat. Presentem *minimum Bayes' risk system combination*, una proposta que automàticament detecta les millors parts de les traduccions candidates i les combina per generar una traducció consensuada que és òptima respecte a una mesura particular de qualitat. Descrivim en profunditat la formalització del nostre enfocament com una suma ponderada de distribucions de probabilitat i proporcionem algoritmes eferents per obtenir la traducció consensuada òptima d'acord amb la molt utilitzada mesura d'avaluació BLEU. Els resultats empírics mostren que el mètode proposat és realment capaç de generar traduccions estadísticament millors que les traduccions candidates proporcionades. En comparació amb altres mètodes de combinació de sistemes de l'estat de l'art, el nostre mètode obté un rendiment semblant i, a mes, no requerix informació addicional llevat de les traduccions candidates.

A continuació, centrem la nostra atenció en com millorar la utilitat de les traduccions automàtiques per a l'usuari final del sistema. Atès que les traduccions automàtiques no són perfectes, una característica desitjable dels sistemes de traducció automàtica és l'habilitat de predir en temps d'execució la qualitat

de les traduccions generades. L'estimació de la qualitat generalment s'aborda com un problema de regressió en que es prediu un valor de qualitat a partir d'un conjunt de característiques que representen la traducció. No obstant això, encara que el concepte de qualitat en la traducció és intuïtivament clar, no hi ha consens sobre quines són les característiques que realment donen compte d'ell. Com a conseqüència, els sistemes d'estimació de la qualitat per traducció han d'utilitzar un gran nombre de característiques febles per predir la qualitat de les traduccions. Això implica diversos problemes d'aprenentatge relacionats amb la colinearidad i ambigüitat de les característiques, i també a causa de la "maledicció" de la dimensionalitat. Nosaltres abordem aquests reptes adoptant una metodologia d'entrenament en dos passos. En primer lloc, un mètode per reduir la dimensionalitat calcula, a partir de les característiques originals, el conjunt de característiques que millor explica la qualitat de les traduccions. A continuació, construïm un model de predicció a partir d'aquest conjunt reduït de característiques per predir finalment el valor de qualitat. Estudiem diversos mètodes de reducció prèviament utilitzats en la literatura i proposem dos nous mètodes basats en el tècniques estadístiques d'anàlisi multivariant. Més específicament, els mètodes proposats per reduir la dimensionalitat es basen en la regressió per mínims quadrats parcials (en anglès, *partial least squares regression*). Els resultats d'una experimentació exhaustiva mostren que els sistemes d'estimació de la qualitat estimats seguint la metodologia en dos passos proposta obtenen una predicció més precisa que els sistemes estimats utilitzant totes les característiques originals. El que és més, un dels mètodes proposats per reduir la dimensionalitat hi va obtenir les prediccions més precises amb només una fracció de les característiques originals. Aquesta ràtio de reducció en el nombre de característiques és important perquè implica una reducció dràstica en els temps d'operació del sistema d'estimació de la qualitat.

Un ús alternatiu dels sistemes de traducció automàtica actuals és incorporarlos a un entorn interactiu d'edició en el qual el sistema i un usuari expert col·laboren per generar traduccions correctes. Aquesta traducció automàtica interactiva ha demostrat que redueix l'esforç de supervisió de l'usuari en comparació amb un sistema de post-edició desacoblat. No obstant això, la traducció automàtica interactiva considera al sistema de traducció un agent passiu en el procés interactiu. En altres paraules, el sistema només suggereix traduccions a l'usuari qui llavors pren les decisions de supervisió necessàries. Com a resultat, l'usuari està obligat a supervisar exhaustivament cada traducció suggerida. Aquesta metodologia passiva assegura l'obtenció de traduccions sense errors però també exigeix un gran esforç de supervisió per part de l'usuari.

Finalment, estudiem diferents tècniques per millorar la productivitat dels sistemes actuals de traducció automàtica interactiva. Específicament, ens centrem en el desenvolupament de metodologies alternatives en que el sistema es converteix en un agent actiu en el procés interactiu. Proposem dues metodologies actives diferents. D'una banda, descrivim una metodologia activa d'interacció en la qual el sistema informa a l'usuari sobre la fiabilitat de les traduccions suggerides. La nostra intuïció és que aquesta informació pugua ajudar l'usuari a localitzar els errors de traducció, millorant per tant la productivitat del sistema. Proposem diferents valors per mesurar la fiabilitat de les traduccions tant a nivell de paraula com a nivell de traducció completa i estudiem la influència que té aquesta informació en la productivitat d'un sistema de traducció automàtica interactiva. Els resultats empírics mostren que el protocol actiu d'interacció proposat és capaç d'aconseguir grans reduccions en l'esforç de supervisió i tot i així generar traduccions de molt alta qualitat. D'altra banda, estudiem un marc d'aprenentatge actiu per a la traducció automàtica interactiva. En aquest cas, el sistema no només és capaç d'informar l'usuari sobre quines traduccions haurien de ser supervisades sinó que també és capaç d'aprendre les traduccions supervisades per l'usuari de manera que millora els seus suggeriments futurs. Desenvolupem un criteri de valor-de-informació per seleccionar les traduccions automàtiques que haurien de ser supervisades per l'usuari. No obstant això, atesa la seua alta complexitat computacional, en la pràctica estudiem diferents estratègies de selecció que aproximen aquest criteri òptim. Els resultats d'una experimentació a gran escala mostren que el marc d'aprenentatge actiu proposat és capaç d'obtenir millors compromisos entre la qualitat de les traduccions generades i l'esforç de supervisió requerit per obtenir-les. El que és més, en comparació amb un sistema de traducció automàtica interactiva, la nostra proposta obté traduccions del doble de qualitat amb el mateix esforç de supervisió.

Contents

Preface

Natural language processing (NLP) is the computerized approach to generate and understand human languages, both oral or written. NLP is part of the artificial intelligence research field, and its origins can be found in the disciplines of linguistics,computer science and cognitive psychology. The goal of NLP is to accomplish human-like language processing for a broad range of tasks or applications, for instance information retrieval, information extraction, question answering, summarization, machine translation, dialog systems, etc.

This thesis explores the area of machine translation (MT), which was the first computer-based application related to natural language. MT investigates the use of computers to translate text (or speech) from a source language into a target language. The first proposals for MT using computers date back to the 1950s. These first attempts, based on information theory, took advantage of the expertise in breaking enemy codes during the second world war and speculated about the underlying principles of natural language. Even after more than 50 years of research, MT remains an open problem.

Different technologies have been proposed in the literature to address MT. These technologies can be classified into two main approaches: rule-based approaches and corpus-based approaches. Rule-based systems uses a set of translation rules created by human translators to generate their output. In contrast, corpus-based systems automatically extract such translation rules from a set of translation examples, also known as corpus or parallel text.

This thesis approaches MT under the statistical framework. Statistical MT (SMT) systems are a type of corpus-based MT systems that use parallel texts to estimate the parameters of a set of statistical models that shape the translation process. Different statistical translation models have been proposed in the literature. Initial SMT models considered the word as the fundamental unit of translation. This simple conception of the translation process does not allow to obtain good translation results due to its inability to capture context information. To solve this problem, a new family of SMT models replaced the words by sequences thereof as the fundamental unit of translation. Among the different multi-word SMT models that have been proposed so far, the so-

called phrase-based models currently constitute the state-of-the-art in SMT. Phrase-based models work by translating sequences of words called phrases. These phrases are not linguistically motivated; instead, they are automatically extracted from corpora using statistical methods.

Despite the success of phrase-based models, current MT systems are still not able to produce ready-to-use translations. Indeed, the output of MT systems usually require human post-editing in order to achieve high-quality translations.This motivates an alternative application of MT system where they collaborate with a human user to generate the final translations. This alternative application receives the name of computer-assisted translation (CAT). CAT is a broad and imprecise term covering a wide range of tools. In this thesis, we sill focus on a specific instantiation of the CAT paradigm which receives the name of interactive MT (IMT). In the IMT framework, the user incrementally generates the desired translation in a series of interactions with the system. This approach allows the system to take advantage of the knowledge of the human translator in contrast to the conventional decoupled post-editing CAT approach.

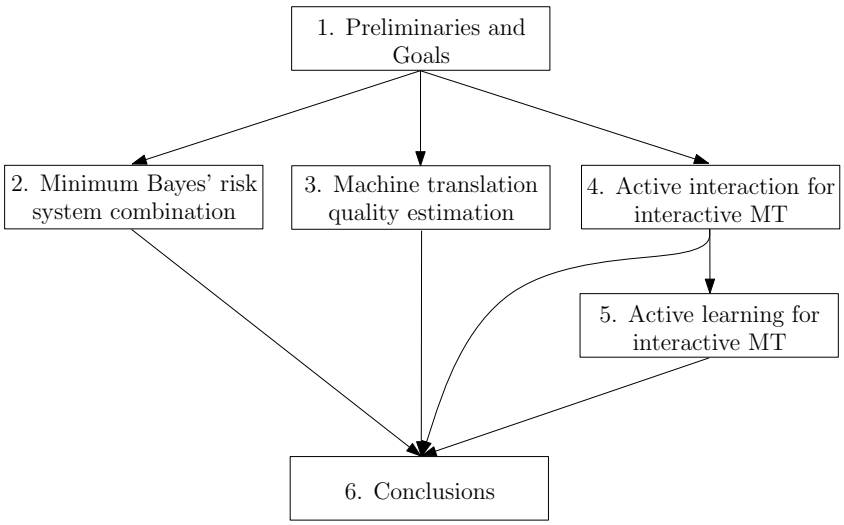
The overall goal of this thesis is to improve the deployment of current MT technology. To accomplish this general goal, we develop contributions in three different research lines:

1. **Combination of MT systems**. As we have said, the output of current MT systems is usually error-prone, however, different systems implementing different MT technologies exhibit different advantages and limitations. Therefore, a proper combination of the output of different systems has the potential to integrate their advantages and get rid of their limitations. So far, system combination approaches for MT either implement sophisticated classifiers to select one of the provided translations, or generate new translations by combining the best subsequences of the provided outputs. We present minimum Bayes' risk system combination (MBRSC), a system combination method for MT that gathers together the advantages of sentence-selection and subsequence-combination methods in a unified multi system minimum Bayes' risk framework. We describe the formal derivation of the model as a weighted ensemble of probability distributions and provide efficient algorithms to obtain the minimum Bayes' risk consensus translation. Regarding the search for the consensus translation, we describe two different formulations for a risk based on the widespread BLEU score, and study different algorithms to obtain the translation of maximum expected BLEU.

2. **Estimation of MT quality**. From the point of view of the end-user, a desirable characteristic to improve the utility of MT systems is the ability to predict at run-time the quality of the generated translations. For instance, this information can help a user to decide if a translation is good for publishing as is. Quality estimation (QE) for MT is usually addressed as a regression problem where a learning model is used to predict a quality score from a (usually highly-redundant) set of features that represent the translation. This redundancy hinders model learning, and thus penalizes the performance of QE systems. We address QE as a two-step regression problem where multiple features are combined to predict a quality score. Given a set of features, we first automatically extract the latent variables that better explain translation quality, and then use them to predict the quality score. We propose different dimensionality reduction methods based on partial least squares regression and compare them against several reduction methods previously used in the QE literature. Moreover, we study how the use of such reduction methods influence the performance of different learning models.

3. **Active protocols for IMT**. IMT systems have shown to reduce the user effort required to supervise automatic translations in comparison to a decoupled post-edition approach. However, IMT considers the translation system as a passive agent in the interaction process. That is, the translation system simply responds to the user interactions. We propose two different IMT approaches where the MT system takes an active part in the interaction with the user. On the one hand, we describe an active interaction approach where the translation system proactively informs the user about the reliability of the suggested translations. We propose various efficient scores to measure translation reliability both at the word and sentence level and study how the availability of such information influences the interaction between the human user and the system. On the other hand, we describe an active learning framework for IMT. In this framework, the user is asked to supervise only a subset of the automatic translations, and the corresponding user-validated translations are used to update the underlying translation model embedded within the IMT system. We propose different measures to decide which automatic translations should be supervised and study how this active learning framework influences the overall user-system translation productivity.

This thesis is structured in six chapters plus a bibliography section. The following figure shows the dependencies between the chapters:



The content of each chapter is as follows:

Chapter 1 introduces the discipline of MT (particularly the SMT framework), the previous approaches presented in the literature to estimate the quality of MT output, and the IMT framework. Next, we present the automatic evaluation measures that were used to empirically evaluate the proposed methods. Finally, we state the general goal of this thesis and the particular goals of each of the research directions explored.

Chapter 2 describes the proposed system combination method, MBRSC. We thoroughly describe the formal derivation of the method and the algorithms proposed to implement it. The chapter ends with a description of the empirical results of the evaluation of MBRSC.

Chapter 3 describes the proposed QE methodology, and the dimensionality reduction methods studied to implement it. Finally, we present the empirical results of a thorough evaluation with multiple different corpus.

Chapter 4 proposes a new active interaction protocol for IMT and describes its application both at the word and sentence levels. The chapter ends with a description of the results obtained in the evaluation of the proposed active interaction protocol.

Chapter 5 proposes a new active learning framework for IMT and describes the different strategies implemented to select which automatic translations should be supervised by the user. Lastly, we present the results of the experimentation carried out to evaluate the proposed active learning framework.

Chapter 6 presents a summary or the work presented in this thesis, including a list of scientific publications, followed by a list of future directions for further developments of the work presented here.

Additionally, we complete the previous content with a set of appendices:

Appendix A describes the practical implementation of IMT using word-graphs. This is the implementation used in the IMT systems described in Chapter 4 and Chapter 5.

Appendix B shows a detailed derivation to compute the free parameters of the linear BLEU definition. The linear BLEU is used in Chapter 2 as an efficient approximation to the exact BLEU risk.

Appendix C describes how to compute n-gram-based feature expectations from word-graphs or translation forests. These expectations are used by the BLEU-based risk function introduced in Chapter 2.

Chapter

1 Preliminaries and Goals

The translation of foreign language texts by computers was one of the first tasks that the pioneers of computing and artificial intelligence set themselves. *Machine translation* (MT) is again becoming an important field of research and development as the need for translations of technical and commercial documentation is growing well beyond the capacity of the translation profession. However, despite intensive research through the last fifty years, MT remains an open problem. Current state-of-the-art MT systems are far from generating error-free translations. Indeed, their translations usually require to be post-edited by human experts in order to be publishable.

This chapter provides an introduction to the general approaches that have been proposed to deal with the MT problem. We then identify the potential drawbacks of current MT technology and examine the challenges and research opportunities available to overcome them. Finally, we describe the research lines explored in this thesis to improve the efficient deployment of current MT technology.

Chapter Outline

1.1 Research scope

Natural language processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. Specifically, the process of a computer extracting meaningful information from natural language input and/or producing natural language output. The complexity of natural language makes NLP an hectic research field that combines theory, methodologies and experts from computer science, linguistics and cognitive psychology. NLP addresses a wide range of challenging applications, including information retrieval, information extraction question-answering, summarizing, machine translation, dialog systems, etc.

This thesis explores the area of machine translation (MT), the NLP discipline that investigates the use of computer software to translate text or speech from one natural language to another.

MT is a fundamental technology that is emerging as a core component of NLP systems. In the multi-lingual society we live in, phenomena such as globalization and technological development have dramatically increase the needs of translation between languages. A good example of multilingualism with high translation needs can be found in the *European Union* (EU) political institutions. The EU is an economic and political confederation of 27 member states and has 23 official languages. Important documents, such as legislation, are translated into every official language. Additionally, the European Parliament[a] provides translation into all languages for documents and its plenary sessions. According to [EC, 2009], the EU employs 1750 translators working full time on translating documents and on other language-related tasks, accompanied by some 600 support staff in management, secretarial, communication, information technology and training functions. To cope with a level of demand that fluctuates in response to political imperatives, the EU used external translation providers which generated approximately one fourth of the EU translation output. The EU also maintained a web translation unit specialized in the translation of web pages. As a result, in 2008 the EU translation services translated more than 1.800.000 pages and spent about one thousand million Euros on translation and interpreting.

Besides being an expensive and time-consuming task, the problem with translation by expert human translators is that, with growing globalization, the demand for high-quality translation has been steadily increasing. Nowadays, there are just not enough qualified translators available to satisfy it. This has dramatically raised the need for improved MT technologies.

[a] http://www.europarl.europa.eu

The idea of MT may be traced back to the 17th century when philosophers such as Leibniz and Descartes put forward theoretical proposals for codes which would relate words between languages. In the 1950s, the Georgetown experiment [IBM, 1954; Hutchins, 2005] involved fully automatic translation of over sixty Russian sentences into English. The experiment was a great success and ushered in an era of substantial funding for MT research. The authors claimed that within three to five years, MT would be a solved problem. However, real progress was much slower. After the ALPAC report [ALPAC, 1966], which found that the ten-year-long research had failed to fulfill expectations, funding was greatly reduced. Beginning in the late 1980s, as computational power increased and became less expensive, more interest was shown in statistical models for MT.

The first proposal for MT using computers was put forward in the 1950s, and was based on the information theory [Shannon, 1948]. It was initiated by a famous publication of Weaver [1955] where the problem of MT was tackled with cryptanalytic techniques inherited from the World War II. This initial intensive research period was followed by a discreet and pragmatic epoch after the ALPAC report. The contributions in the statistical MT field were minor until the early nineties, when the IBM group presented the Candide system [Berger et al., 1994], a statistical MT system [Brown et al., 1990, 1993] that was demonstrated to be competitive with state-of-the-art rule-based systems built from expert linguistic knowledge. Since then, the development of statistical MT has experienced a major boost that seems to have reached a technical plateau nowadays [Lopez, 2008]. Despite the intensive research, it seems that many experts in the area agree that the performance of MT technology after more than fifty years of developments leaves much to be desired [NIST, 2006; Callison-Burch et al., 2012]; fully-automatic high-quality MT remains an open problem.

In this thesis, we explore three different research lines to improve the broader deployment of current error-prone MT technology. First, we focus on the improvement of fully-automatic MT technology by proposing a system combination approach that combines the outputs of different MT systems into a new improved consensus translation. Then, we propose a new methodology to estimate at run-time the quality of the translations automatically generated so we can improve their utility for the end-user. For example, by informing the user of unreliable translations that should be revised before publication. Finally, we focus on the computer-assisted translation technology. Specifically, we study different approaches to improve the user-machine interaction with the objective of boosting the productivity of such systems.

The rest of this chapter is organized as follows. In Section 1.2, we present a classification of the different strategies and technologies that have been historically applied to tackle the MT problem. In Section 1.3, we further describe in detail the statistical approach to MT since it is the approach on which this thesis is focused. In section 1.4, we formalize the quality estimation task and further describe some of its application to MT. In Section 1.5, we formalize the interactive MT approach and present some details of how it is implemented in practice. In Section 1.6, we describe the different assessment measures typically used to evaluate translation quality, user translation-supervision effort, and quality estimation accuracy. In Section 1.7 we present the research lines explored in this thesis to improve the practical deployment of current MT technology and the specific goals pursued to accomplish the general goal of the thesis. Finally, we summarize the contents of this chapter in Section 1.8.

1.2 Classification of MT Systems

The great variety of MT approaches proposed in the literature can be distinguished according to different criteria. We distinguish the type of the input provided to the system, the application for which the system is used, the level of analysis, and the core technology used.

1.2.1 By Type of Input

Most MT systems deal with text input. In this case, the input text can typically be expected to be grammatically correct and well-formed. More complex is the case of speech translation [Vidal, 1997; Ney, 1999; Casacuberta et al., 2004] where the system has to deal with speech recognition errors and spontaneous speech phenomena such as ungrammatical utterances, false starts or hesitations. In this thesis, we focus on translation of text input.

1.2.2 By Application

There are various types of applications for MT technology. In *gisting*, the goal is to decide whether a text in a foreign language contains relevant information. Typically a human translation would then be performed to extract this information. In *post-editing* applications, the aim is to produce an approximated translation of the input that will be corrected by a human translator in a separated step. In *interactive* applications, a human translator and an MT system collaborate to generate the translation. The MT system produces translations that are amended by the user. Whenever a translation is wrong,

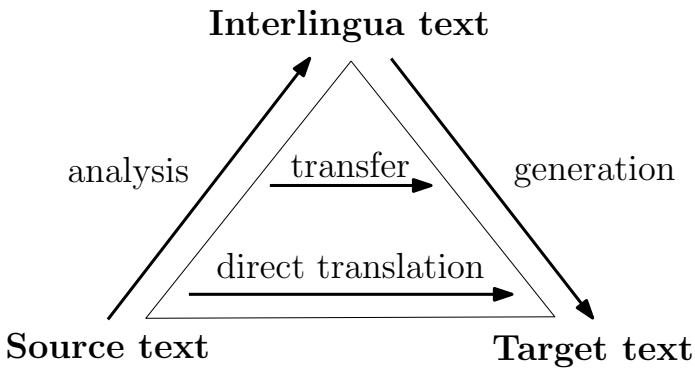


Figure 1.1: Bernard Vauquois' pyramid showing comparative depths of intermediary representation, interlingual machine translation at the peak, followed by transfer-based, then direct translation.

the system considers the feedback of the user to perform a new translation. Such process is repeated until the provided translation matches the user's expectations. Finally, in *fully automatic* applications, the computer is used to directly produce final translations. Using state-of-the-art technology, high quality translations can only be produced for very restricted domains like weather forecast [Langlais et al., 2005] or hotel reception questions [Amengual et al., 2000].

1.2.3 By Level of Analysis

We can distinguish three different types of MT systems according to the level of analysis of the source sentence before translating: direct translation, transfer approach and interlingua approach. Figure 1.1 shows the Vauquois' pyramid [Vauquois, 1975], the standard visualization of these three approaches.

Direct Translation

This is the simplest approach and thus it was adopted by the first MT systems. The direct approach performs a word-by-word translation from the source language into the target language. It can include a morphosyntactic analysis of the source text to capture grammar categories and other morphological information, but it typically excludes relationships between groups of words.

Transfer Approach

The transfer approach divides the translation process intro three steps: analysis, transfer, and generation. In the analysis step, the input text is syntactically and semantically analyzed to produce an abstract representation of the source sentence. In the transfer step, this representation is transferred into a corresponding representation in the target language. In the generation step, a target language sentence is produced from this target language representation.

Interlingua Approach

In the interlingua approach, a deep fine-grained analysis is performed to obtain a representation of the input text in an abstract language-independent representation, the so-called interlingua. Then, the interlingua representation is used to produce the target language sentence. In other words, the input text is first understood, and then translated. The main advantage of the interlingua over other MT approaches is that the development of translation systems between all pairs of a set of n languages is more efficient. Only n systems are needed, one system to translate between each language and interlingua. In contrast the transfer or the direct translation approaches requires $n \cdot (n-1)$ systems, one system between each pair of languages.

1.2.4 By Core Technology

The vast majority of introductory works on MT [Hutchins and Somers, 1992; Trujillo, 1999; Lopez, 2008] classify MT systems depending on the translation technology that they use. We can identify two main approaches: rule-based systems, and corpus-based systems. Nevertheless, despite using opposite technologies, a number of proposals that combine both approaches can be found in the literature [Chen and Chen, 1996; Lagarda et al., 2009].

Rule-based Approaches

Rule-based systems are characterized by a set of rules which are aimed at describing the translation process. Typically, these rules that depend on the specific source and target languages are specified manually by a human expert. This is a very slow and expensive process for which linguistic experts are needed. Rule-based systems were the predominant MT approach during the 1970s and 1980s, as expemplified by the Systran system [Toma, 1977]. However, due to their high cost, nowadays are mainly used to translate for very specific domains. For example, PAHOMTS [Aymerich and Camelo, 2009] is

a rule-based MT system is specialized in translating text of the medical and pharmaceutical domains.

Corpus-based Approaches

Under this empirical approach, the knowledge sources to develop MT systems are computed automatically by analyzing example translations (also known as parallel texts, or translation corpora). The main advantage of the corpus-based approach is that it allow for a very quick development of MT systems for new language pairs and/or new domains under the assumption that a suitable amount of training data is available. Corpus-based approaches can be classified into two groups: example-based approaches, and statistical MT approaches.

- **Example-based approach:** Also known as *memory-based* approach, it is characterized by its use of translation examples as its main knowledge base at run-time [Somers, 1999, 2003]. Example-based MT rejects the idea that people translate by doing deep linguistic analysis. Instead it is founded on the belief that people translate firstly by decomposing a sentence into certain segments, then by translating these segments, and finally by properly composing these fragments into one long sentence. Translation memories [Kay, 1998] are a popular approach to implement example-based MT. TRADOS [SDL, 2013] and Déjà Vu [Atril, 2013] are two commercial MT systems based on translation memories.

- **Statistical approach:** Statistical MT (SMT) systems generate the translations on the basis of statistical models whose parameters are derived from the analysis of the parallel corpora. SMT can be implemented in a number of ways including well-known machine learning approaches such as neuronal networks [Castaño and Casacuberta, 1997], finite state transducers [Knight and Al-Onaizan, 1998; Casacuberta and Vidal, 2004], and structured prediction methods [Liang et al., 2006]. In comparison with other MT approaches, SMT is mathematically well-founded, non language-dependent, efficient, and allows for a fast development of MT systems provided that parallel corpora is available. These are some of the reasons why SMT is the most widely-studied MT approach nowadays. Next section provides a formal description of the SMT approach.

1.3 Statistical Machine Translation

We now review the statistical pattern recognition approach to the translation problem, namely the *statistical machine translation* (SMT) approach. SMT formalizes MT as a decision problem where it is necessary to decide upon a sequence of target language words given a sequence of source language words. SMT considers every sentence in the target language as a possible translation of any source sentence, and uses the Bayesian decision theory [Duda et al., 2001] to select the correct translation. Thus, we first describe the Bayesian decision theory in Section 1.3.1, and then, we present the two main SMT formulations of the translation problem: the classical source-channel model in Section 1.3.2, and the direct maximum entropy model in Section 1.3.3.

1.3.1 Decision Theory

Bayesian decision theory is a fundamental statistical approach that quantifies the trade-off between various decisions using probabilities and costs that accompany such decisions. Statistical pattern recognition uses the decision theory to solve many practical classification problems. A classification problem is stated as the problem of choosing which class a given object belongs to. Let $\mathcal{X}$ be the domain of the objects that a classification system might observe; and $\mathcal{Y}$ the set of classes $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{|\mathcal{Y}|})$. Then, a classification system is characterized by a function that maps each object to one class, the so-called classification function $\mathrm{C} : \mathcal{X} \to \mathcal{Y}$ [Duda et al., 2001].

The performance of a classification function is usually measured as a function of the classification error. However, there are problems in which all the classification errors do not have the same repercussions. Therefore, a function that ranks these mistakes should be provided. The loss function, $\mathrm{L}(\mathbf{y}, \mathbf{y}')$, evaluates the loss that is incurred by the classification function when classifying the object $\mathbf{x}$ in to the class $\mathbf{y}$, knowing that the correct class is $\mathbf{y}'$.

Within this framework, the performance of a given classifier can be measured by its global risk that characterizes the contribution of all objects in the performance of the classifier. The global risk $\mathrm{R}(\mathrm{C})$ of a classifier C is formally defined as follows[b]:

$$\mathrm{R}(\mathrm{C}) = \mathbb{E}_{\mathcal{X}}\left[\mathrm{R}(\mathrm{C}(\mathbf{x}) \mid \mathbf{x})\right] = \int_{\mathcal{X}} \mathrm{R}(\mathrm{C}(\mathbf{x}) \mid \mathbf{x}) \cdot \Pr(\mathbf{x}) \cdot d\,\mathbf{x} \tag{1.3.1}$$

[b]The notation convention will be as follows. The symbol $\Pr(\cdot)$ is used to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, the generic symbol $\mathrm{P}(\cdot)$ is used.

where $\mathrm{R}(\mathrm{C}(\mathbf{x}) \mid \mathbf{x})$ is the conditional risk given $\mathbf{x}$, i.e. the expected loss of classifying $\mathbf{x}$ in the class $\mathbf{y} = \mathrm{C}(\mathbf{x})$ determined by the classifier C. This conditional risk is expressed as follows:

$$\mathrm{R}(\mathrm{C}(\mathbf{x}) \mid \mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \mathrm{L}(\mathrm{C}(\mathbf{x}), \mathbf{y}') \cdot \Pr(\mathbf{y}' \mid \mathbf{x}) \tag{1.3.2}$$

In practice however, calculating the global risk when comparing systems requires the classification of all possible objects. Therefore, an empirical risk on a test set $\mathcal{T}$ is usually computed instead of the global risk:

$$\overline{\mathrm{R}}_{\mathcal{T}}(\mathrm{C}) = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{x} \in \mathcal{T}} \mathrm{R}(\mathrm{C}(\mathbf{x}) \mid \mathbf{x}) \tag{1.3.3}$$

Minimizing the conditional risk for each object $\mathbf{x}$ is a sufficient condition to minimize the global risk. Therefore, the optimal classification rule for loss function $\mathrm{L}(\mathbf{y}, \mathbf{y}')$, namely the *minimum Bayes' risk* (MBR) classifier, is the one that minimizes the conditional risk for each object [Duda et al., 2001]:

$$\hat{\mathbf{y}} = \widehat{\mathrm{C}}(\mathbf{x}) = \arg\min_{\mathbf{y} \in \mathcal{Y}} \mathrm{R}(\mathbf{y} \mid \mathbf{x}) = \arg\min_{\mathbf{y} \in \mathcal{Y}} \sum_{\mathbf{y}' \in \mathcal{Y}} \mathrm{L}(\mathbf{y}, \mathbf{y}') \cdot \Pr(\mathbf{y}' \mid \mathbf{x}) \tag{1.3.4}$$

where $\mathbf{x}$ is the object to be classified, and $\hat{\mathbf{y}}$ is the class selected by the MBR classifier $\widehat{\mathrm{C}}(\mathbf{x})$. Depending on the loss function of interest, there exist different optimal classification rules.

A common practical approach is to consider that each classification error has the same importance. This is can be done by assuming a $0-1$ loss function which only distinguishes two sorts of actions: wrong classification (loss of 1) and correct classification (zero loss):

$$\mathrm{L}_{0-1}(\mathbf{y}, \mathbf{y}') = \begin{cases} 0 & \mathbf{y} = \mathbf{y}' \\ 1 & \text{otherwise} \end{cases} \tag{1.3.5}$$

If we consider the 0-1 loss function, the minimum Bayes' risk classifier in Equation (1.3.4) can be greatly simplified. This approach is known in the literature as the optimal Bayes' classification rule [Duda et al., 2001]:

$$\hat{\mathbf{y}} = \mathrm{C}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Pr(\mathbf{y} \mid \mathbf{x}) \tag{1.3.6}$$

This is the formulation commonly followed to develop MT systems.

1.3.2 Source-channel Model

Following the decision theory, translation can be formalized as follows. The objects that the classification system might observe are now sequences of words (sentences) in a source language; and the set of classes is the set of all possible sequences of words (translations) in a target language. Given a source sentence $\mathbf{f}$ in the source language $\mathcal{F}$, the goal is to obtain its equivalent translation $\mathbf{e}$ in the target language $\mathcal{E}$. From the set of all possible target language sentences, we are interested in that $\hat{\mathbf{e}}$ with the highest probability:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathcal{E}} \Pr(\mathbf{e} \mid \mathbf{f}) \tag{1.3.7}$$

However, $\Pr(\mathbf{e} \mid \mathbf{f})$ is usually difficult to estimate so the Bayes' rule [Bayes, 1763] is usually applied to achieve the following decomposition:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathcal{E}} \frac{\Pr(\mathbf{e}) \cdot \Pr(\mathbf{f} \mid \mathbf{e})}{\Pr(\mathbf{f})} = \arg\max_{\mathbf{e}\in\mathcal{E}} \Pr(\mathbf{e}) \cdot \Pr(\mathbf{f} \mid \mathbf{e}) \tag{1.3.8}$$

where the term $\Pr(\mathbf{f})$ has been dropped since it does not depend on the maximization variable $\mathbf{e}$.

Equation (1.3.8) is known as the source-channel model for SMT [Brown et al., 1990], and sometimes it is also referred as the "fundamental equation of SMT" [Brown et al., 1993]. Those SMT models that implement this source-channel approach are usually referred to as generative models. Here, the term $\Pr(\mathbf{e} \mid \mathbf{f})$ has been decomposed into a language model $\Pr(\mathbf{e})$ and a translation model $\Pr(\mathbf{f} \mid \mathbf{e})$. Intuitively, the translation model models the correlation between the source and target sentences, but can be also be understood as a mapping between source and target words. The language model on the other hand measures the well-formedness of the candidate translation. Typically, Equation (1.3.8) is favored over the direct translation model of Equation (1.3.7) with the argument that it yields a modular approach. Instead of modeling one probability distribution, we model two different knowledge sources that can be trained independently.

Typically, training is performed by applying the well-known maximum likelihood approach. If the language model $\Pr(\mathbf{e}) \approx \mathrm{P}_{\boldsymbol{\gamma}}(\mathbf{e})$ depends on parameters $\boldsymbol{\gamma}$, and the translation model $\Pr(\mathbf{f} \mid \mathbf{e}) \approx \mathrm{P}_{\boldsymbol{\theta}}(\mathbf{f} \mid \mathbf{e})$ depends on parameters $\boldsymbol{\theta}$, then the optimal parameter values are obtained by maximizing the likelihood

on a parallel training corpus $\{(\mathbf{f}_n, \mathbf{e}_n)\}_{n=1}^{N}$ [Brown et al., 1993]:

$$\widehat{\boldsymbol{\gamma}} = \arg\max_{\boldsymbol{\gamma}} \prod_{n=1}^{N} \mathrm{P}_{\boldsymbol{\gamma}}(\mathbf{e}_n) \tag{1.3.9}$$

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N} \mathrm{P}_{\boldsymbol{\theta}}(\mathbf{f}_n \mid \mathbf{e}_n) \tag{1.3.10}$$

Various SMT systems were developed following this approach [Brown et al., 1993; Vogel et al., 1996]. Yet, among other problems [Och and Ney, 2002], the decision rule stated in Equation (1.3.8) is optimal under the assumption of a $0-1$ loss function. This loss function is better known in SMT as the *sentence error rate* (SER), and it is a particularly inadequate performance measure. SER considers that there is an error if the translation given by the system is not identical to the reference translation. In other words, SER considers a translation to be completely erroneous even though it differs in only one word from the reference. SER provides a rough and superficial evaluation of the translation quality and is rarely used in favor of other more sophisticated evaluation measures. Two excellent discussions on the use of different loss functions in SMT can be found in [Andrés-Ferrer et al., 2008; Schlueter et al., 2012].

1.3.3 Maximum Entropy Model

An alternative to the source-channel approach is to directly model the posterior probability $\Pr(\mathbf{e} \mid \mathbf{f})$ in Equation (1.3.7) using the maximum entropy framework [Berger et al., 1996; Papineni et al., 1998; Och and Ney, 2002]. The SMT models that implement this approach are usually referred as *log-linear models*[c]. Log-linear models are characterized by a set of feature functions $h_m(\mathbf{e}, \mathbf{f})$ and a corresponding set of free parameters λ_m. Formally, the direct translation probability is modeled as follows:

$$\Pr(\mathbf{e} \mid \mathbf{f}) \approx \mathrm{P}_{\boldsymbol{\lambda}}(\mathbf{e} \mid \mathbf{f}) = \frac{\exp\left(\sum_m \lambda_m h_m(\mathbf{e}, \mathbf{f})\right)}{\sum_{\mathbf{e}' \in \mathcal{E}} \exp\left(\sum_m \lambda_m h_m(\mathbf{e}', \mathbf{f})\right)} \tag{1.3.11}$$

[c] Also known as maximum-entropy models, exponential models, and Gibbs models.

Similarly as done in the source-channel approach, we can ignore the denominator during the search process because it does not depend on the hypothesized translation. The final decision rule is then stated as follows:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathcal{E}} \sum_{m} \lambda_m h_m(\mathbf{e} \mid \mathbf{f}) \tag{1.3.12}$$

Note the source-channel approach in Equation (1.3.8) is a special case of the maximum entropy approach where only the following two feature functions are used and their weights are set to one:

$$h_1(\mathbf{e}, \mathbf{f}) = \log(\mathrm{P}_{\widehat{\boldsymbol{\gamma}}}(\mathbf{e})) \tag{1.3.13}$$

$$h_2(\mathbf{e}, \mathbf{f}) = \log(\mathrm{P}_{\widehat{\boldsymbol{\theta}}}(\mathbf{f} \mid \mathbf{e})) \tag{1.3.14}$$

We can again use the maximum likelihood criterion to optimize the parameters λ_m in Equation (1.3.12):

$$\widehat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \prod_{n=1}^{N} \mathrm{P}_{\boldsymbol{\lambda}}(\mathbf{e}_n, \mathbf{f}_n) \tag{1.3.15}$$

This maximization problem can be solved by using the *generalized iterative scaling* (GIS) algorithm [Darroch and Ratcliff, 1972]. However, the application of the GIS algorithm is very costly due to the necessity of computing the normalization factor present in Equation (1.3.11).

Alternatively, the maximum likelihood criterion can be replaced by a criterion based on automatic evaluation methods. In this case, we assume that the best model is the one that produces the smallest overall error with respect to a given error function. This new optimization approach is known as *minimum error rate training* (MERT) algorithm [Och, 2003]. MERT can be implemented by means of different optimization algorithms: Och [2003] proposed the use of the Powell's conjugate gradient descent method [Powell, 1964], but alternative algorithms such as the downhill-simplex algorithm [Nelder and Mead, 1965] or the *margin infused relaxed algorithm* (MIRA) [Crammer and Singer, 2003] can also be used. Cherry and Foster [2012] provide a thorough comparison of several parameter-tuning strategies for maximum entropy SMT models.

1.4 Estimating the Quality of MT Outputs

Although significant progress has been observed in the overall quality of MT technology in recent years, fully-automatic MT systems are not robust enough

and the quality of the generated translations can vary considerably across translation segments. Thus, the capability of predicting the reliability of the generated translations is a desirable feature of currently error-prone MT technology. The estimation of the quality of MT translations is particularly appealing when we consider the end-user of the MT system. In this context, quality estimates can help the user to get the most of MT technology in a number of scenarios, for example:

- A professional translator can use an estimation of the quality to decide if a translation is worth to be post-edited, or it will cost more to post-edit the translation than translate it from scratch [Specia et al., 2009a].
- If source sentences are not available, or if the user is not fluent in the source language, quality estimates can be used to inform the user about the quality of the translations, e.g. to highlight certain translations as "not reliable" [Blatz et al., 2004; Specia et al., 2009b].
- If multiple translations for a given source sentence are available, quality estimates can help to decide which translation is the "best" and, thus, should be selected [Nomoto, 2004; Ueffing and Ney, 2005].

Historically, translation quality assessment has been done manually by human experts. These experts need to read the automatic translation and the source text to be able to judge whether the translation is good or not which, obviously, is a very time consuming task particularly for long sentences. Moreover, in some cases it may not be even possible to assess translation quality. If the human expert do not speak the source language, he may be able to judge the fluency of the translation but can not say anything about its adequacy. Therefore, automatic quality assessment of MT translations is a crucial problem for the practical deployment of MT technology.

This task, referred to as confidence or *quality estimation* (QE), is concerned about predicting MT output quality without any information about the expected output. We distinguish the task of QE from that of MT evaluation (further discussed in Section 1.6.1) by the need in the latter of reference translations. The goal of MT evaluation is to compare an automatic translation to one (or several) reference translation(s) and provide a quality score which reflects how close the two translations are. In QE, the task consists in estimating the quality of the translation given only the source sentence, the translation, and, possibly, some information about the translation process.

The different QE approaches proposed in the literature can be distinguished according to different criteria. According to the translation element for

which quality information is computed, we can distinguish between word-level QE [Gandrabur and Foster, 2003; Ueffing and Ney, 2007; Sanchis et al., 2007], sentence-level QE [Blatz et al., 2004; Quirk, 2004; Gamon et al., 2005; Specia et al., 2009b], or document-level QE [Soricut and Echihabi, 2010]. Depending on the type of the quality score to be predicted, we can distinguish between approaches that predict a probability of correction, i.e. a binary score [Blatz et al., 2004; Quirk, 2004; Gamon et al., 2005; Ueffing and Ney, 2007; Sanchis et al., 2007], and approaches that estimate a continuous quality score [Specia et al., 2009b]. Finally, depending on the estimation model, we can distinguish between QE approaches that compute a direct estimation of the quality by means of a single indicator function, and systems that use a machine learning model to predict the quality score from several indicators, namely features, that represent the translation.

Nowadays, QE is typically addressed as a regression problem. Let $\mathbf{x}$ be a feature vector representing the translation, and y be the quality score associated to the translation. The features in $\mathbf{x}$ may capture different aspects of the translation. In general, features may depend on the source sentence and / or the translation, but should not depend, for example, on the reference translation which would be unavailable at testing time. The basic approach is to define a function, namely a parametrized model, $\mathbb{M}(\mathbf{x};\boldsymbol{\theta})$ (where $\boldsymbol{\theta}$ is a parameter vector) intended to be correlated with the corresponding quality score y. Since the way in which y and $\mathbf{x}$ actually relate is usually unknown, $\mathbb{M}(\cdot;\boldsymbol{\theta})$ is instantiated to different flexible models, such as support vector machines [Cortes and Vapnik, 1995], whose free parameters $\boldsymbol{\theta}$ can be estimated to fit a given data set $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$. This point of view provides a solid framework where to derive accurate frameworks. In exchange, usually a large amount of effort is usually required from translation experts to define adequate features.

1.5 Interactive Machine Translation

The application of statistical pattern recognition techniques to the field MT has allowed the development of new MT systems with less effort than it was previously required under the formerly dominant rule-based paradigm [Koehn, 2010]. However, the quality of the translations produced by any fully-automatic (statistical, memory-based or rule-based) MT system still remain below than that of human translation. This quality could be enough for many applications, but for others, the output of the MT systems has to be revised by a human expert to reach publishable level. This approach, where computer software supports and facilitates the translation process of a human user is known

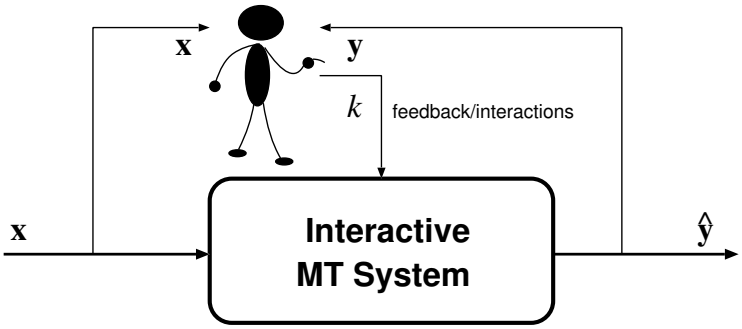


Figure 1.2: Diagram of an interactive MT system. To translate a source sentence $\mathbf{x}$, the user interacts with the system accepting or correcting the proposed translations $\mathbf{y}$. User feedback k is used by the system to improve its suggestions.

as *computer assisted translation* (CAT) [Isabelle and Church, 1998].

Nowadays, typical CAT approaches implement a serial process in which an MT system provides complete translations which are then corrected (post-edited) by a human expert. It should be noted that there is no actual interaction between the MT system and the translator in this scenario, since they work as two isolated processes. This serial work-flow is the main drawback of the post-editing CAT approach, since it prevents MT systems from taking advantage of the knowledge of the human translators, and human translators cannot take advantage of the adaptive ability of MT systems.

An alternative to this serial post-editing CAT process is the interactive CAT approach proposed in the TransType [Foster et al., 1998; Langlais et al., 2000; Langlais and Lapalme, 2002] and TransType2 [Casacuberta et al., 2009; Barrachina et al., 2009] projects. In the interactive CAT approach, a fully-fledged MT engine is embedded into an interactive editing environment, and used to generate suggested completions of each target sentence being translated. These completions may be accepted or amended by the human translator. Once validated, they are exploited by the MT engine to produce further, hopefully improved suggestions. This new approach, schematized in Figure 1.2, is known as *interactive machine translation* (IMT). TransType allowed only single-token completions, where a token could be either a word or a short sequence of words from a predefined set of sequences. This idea was extended to full target sentence completions in the TransType2 project. IMT has significant potential advantages over traditional post-editing where there is no way for the system to benefit from the corrections of the user.

Interactivity in MT and CAT has been explored for a long time to solve different types of ambiguities [Barrachina et al., 2009]. However, there are only few research groups that have published, to our knowledge, contributions in the IMT topic. As we have mentioned, the first publications are related with the TransType project [Foster et al., 1998; Langlais et al., 2000; Foster, 2002; Langlais and Lapalme, 2002; Nepveu et al., 2004; Patry and Langlais, 2009; Simard and Isabelle, 2009]. The second group of publications are around the TransType2 project [Cubel et al., 2003; Och et al., 2003b; Civera et al., 2004a,b; Cubel et al., 2004; Bender et al., 2005; Tomás and Casacuberta, 2006; Barrachina et al., 2009; Casacuberta et al., 2009]. More recently other research groups have started to work on this topic [Koehn and Haddow, 2009].

A problem inherent to any interactive system is that the human user may choose between many ways or "actions" to provide the interaction feedback. Obviously, in order to allow for a proper implementation of the IMT approach, human creativity has to be limited in some way so that the system can take maximum advantage of the allowed interactive actions. This kind of limitation of user actions is often referred to as *user model* [Fischer, 2001] in the human-computer interaction literature. In the rest of this thesis, we will interchangeably use user model and the more modest term *interaction protocol*.

Figure 1.3 displays a typical IMT session that exemplifies the IMT interaction protocol. Let us suppose that a source Spanish sentence $\mathbf{f}$ ="Transferir documentos explorados a otro directorio" is to be translated into a target English sentence $\hat{\mathbf{e}}$. Initially, with no user feedback, the system suggests a complete translation $\mathbf{e}_s$ ="Move documents scanned to other directory". From this translation, the user marks a prefix $\mathbf{e}_p$ ="Move " as correct and begins to type the rest of the target sentence. Depending on the system or the user's preferences, the user might type the full next word, or only some letters of it (in our example, the user types the single next character "s"). Then, the MT system suggests a new suffix $\mathbf{e}_s$ ="canned documents to other directory" that completes the validated prefix and the input the user has just typed ($\mathbf{e}_p$ ="Move s"). The interaction continues with a new prefix validation followed, if necessary, by new input from the user, and so on, until the user considers the translation to be complete and satisfactory.

We can formalize this interaction process as a classification problem similarly as we have shown for fully-automatic translation in Section 1.3. In this case, we must decide upon a suffix $\mathbf{e}_s$ that completes a prefix $\mathbf{e}_p$ validated by the user to obtain a complete translation of the source sentence $\mathbf{f}$:

$$\hat{\mathbf{e}}_s = \arg\max_{\mathbf{e}_s} \Pr(\mathbf{e}_s \mid \mathbf{f}, \mathbf{e}_p) \tag{1.5.1}$$

source (**f**): Transferir documentos explorados a otro directorio
desired translation ($\hat{\mathbf{e}}$): Move scanned documents to another folder

interaction-0	$\mathbf{e}_p$	
	$\mathbf{e}_s$	*Move documents scanned to other directory*
interaction-1	$\mathbf{e}_p$	Move
	k	**s**
	$\mathbf{e}_s$	*canned documents to other directory*
interaction-2	$\mathbf{e}_p$	Move scanned documents to
	k	**a**
	$\mathbf{e}_s$	*nother directory*
interaction-3	$\mathbf{e}_p$	Move scanned documents to another
	k	**f**
	$\mathbf{e}_s$	*older*
accept	$\mathbf{e}_p$	Move scanned documents to another folder

Figure 1.3: IMT session to translate a Spanish sentence into English. The desired translation is the translation the human user wants to obtain. At interaction-0, the system suggests a translation ($\mathbf{e}_s$). At interaction-1, the user moves the mouse to accept the first five characters "Move " and presses the **s** key (k), then the system suggests completing the sentence with "*scanned documents to other directory*" (a new $\mathbf{e}_s$). Interactions 2 and 3 are similar. In the final interaction, the user accepts the current translation.

which can be straightforwardly rewritten as:

$$\begin{aligned}\hat{\mathbf{e}}_s = \arg\max_{\mathbf{e}_s} \Pr(\mathbf{e}_s \mid \mathbf{f}, \mathbf{e}_p) &= \arg\max_{\mathbf{e}_s} \frac{\Pr(\mathbf{e}_p, \mathbf{e}_s \mid \mathbf{f})}{\Pr(\mathbf{e}_p \mid \mathbf{f})} \\ &= \arg\max_{\mathbf{e}_s} \Pr(\mathbf{e}_p, \mathbf{e}_s \mid \mathbf{f}) \qquad (1.5.2)\end{aligned}$$

where the term $\Pr(\mathbf{e}_p \mid \mathbf{f})$ can be ignored since it does not participate in the maximization $\arg\max_{\mathbf{e}_s}$.

Given that $\mathbf{e}_p\,\mathbf{e}_s = \mathbf{e}$, this equation is very similar to the SMT formalization in Equation (1.3.7). The main difference is that the search now is performed over the set of suffixes $\mathbf{e}_s$ that complete the prefix $\mathbf{e}_p$ provided by the user instead of over the set of target language sentences. This implies that we can use the same SMT models whenever the search procedures are adequately modified [Och et al., 2003b]. It should be noted that SMT models are usually defined at word level while the IMT interface can be configured to work at

character level as in Figure 1.3. This is not an important issue since the transformations that are required in the SMT models for their use at character level are trivial.

The IMT optimization problem in Equation (1.5.2) is thus reduced to a search problem constrained by the prefix. Obviously, there can be other alternatives, but this one has the advantage that we can use the same models as for SMT, and therefore, we can also use the same training algorithms [Barrachina et al., 2009]. Regarding the IMT search, it can be carried out by a modification of the available search algorithms for SMT [Barrachina et al., 2009]. A key aspect to take into account is the speed of the search process. Typically system suggestions must be produced in real time after each user keystroke [Och et al., 2003b; Barrachina et al., 2009]. Thus, instead of a full decoding after each interaction, suggestions are searched in a previously-generated word-graph that represents a large set of possible translations of the source sentence. Specifically, the system finds the best path in the word-graph which is compatible with the user prefix.

A word-graph is a weighted directed acyclic graph in which each node represents a partial translation hypothesis and each edge is labeled with a word (or group of words) of the target sentence and is weighted according to the scores given by an SMT model. Word-graphs can be easily generated as a by-product of the translation process [Ueffing et al., 2002; Koehn, 2003; Hasan et al., 2007]. The main advantages of word-graph-based IMT systems is their efficiency in terms of the time cost per each interaction. This is due to the fact that the word graph is generated only once at the beginning of the interactive translation process of a given source sentence, and the suffixes required in IMT can be obtained by incrementally processing this word-graph.

A problem arises when the user sets a prefix which cannot be explained by the statistical models. Under these circumstances, the suffix cannot be appropriately generated since no path in the word-graph is compatible with the user prefix. The common procedure to address this problem is to perform a tolerant search in the word-graph. This smoothed search uses the well known concept of Levenshtein distance in order to obtain the most similar string for the given prefix. Further details of the use of word-graph to efficiently implement practical IMT systems are given in Appendix A.

1.6 Assessment Criteria

We now describe the assessment measures that that will be used to evaluate the soundness of the techniques, methods, and strategies proposed in this thesis.

As described in the previous section, we are focused on MT, and particularly, on the efficient deployment of MT technology. In other words, our goal is not only to develop MT systems that provide translations of the highest quality, but we also take into account the human effort required to develop and deploy those systems. Therefore, the assessment measures are twofold: the quality of the translations generated, and the human effort involved in the generation of those translations. Additionally, we also describe the methodology followed to determine the statistical significance of the empirical results.

1.6.1 Translation Quality

Although the criteria that should be taken into account in assessing translation quality is fairly intuitive and well established, translation evaluation is still a complex and task dependent task. That is the reason why translation quality evaluation has traditionally been performed by human experts. However, automatic translation quality evaluation has many advantages over manual evaluation: it is faster, easier, and cheaper. Moreover automatic evaluation metrics can be applied on a frequent and ongoing basis during system development. This allows the designers to guide the development of the system based on concrete performance improvements.

Several methods have been proposed in recent years to automatically evaluate MT quality by comparing candidate translations with reference translations. Examples of such methods are word error rate [Levenshtein, 1966], position-independent word error rate [Amengual et al., 2000; Casacuberta et al., 2004], generation string accuracy [Bangalore et al., 2000], multi-reference word error rate [Nießen et al., 2000], BLEU score [Papineni et al., 2002], NIST score [Doddington, 2002], METEO [Banerjee and Lavie, 2005], and TER [Snover et al., 2006]. Through this thesis, we will use the widespread BLEU and TER measures to evaluate translation quality.

Bilingual Evaluation Understudy (BLEU)

BLEU [Papineni et al., 2002] was one of the first automatic measures to achieve a high correlation with human judgments of quality [Coughlin, 2003], and remains one of the most popular automated and inexpensive measures. It computes a value between zero and one that indicates to which extent the candidate translation contains the same information as the reference translation. This value is usually interpreted as a percentage where a value equal to 100% denotes a candidate translation equal to the reference. Formally, BLEU

computes the geometric average of the precision, $\mathrm{PR}_n(\mathbf{e}, \mathbf{e}')$, of n-grams[d] of various lengths (typically up to four[e]) between the candidate $\mathbf{e}$ and the reference translation $\mathbf{e}'$. This average is multiplied by a factor, namely the brevity penalty $\mathrm{BP}(\mathbf{e}, \mathbf{e}')$, that penalizes translations shorter than the reference:

$$\mathrm{BLEU}(\mathbf{e}, \mathbf{e}') = \left(\prod_{n=1}^{4} \mathrm{PR}_n(\mathbf{e}, \mathbf{e}') \right)^{\frac{1}{4}} \cdot \mathrm{BP}(\mathbf{e}, \mathbf{e}') \tag{1.6.1}$$

The n-gram precisions and the brevity penalty are computed as:

$$\mathrm{PR}_n(\mathbf{e}, \mathbf{e}') = \frac{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{e})} \min(\#_{\mathbf{w}}(\mathbf{e}), \#_{\mathbf{w}}(\mathbf{e}'))}{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{e})} \#_{\mathbf{w}}(\mathbf{e})} \tag{1.6.2}$$

$$\mathrm{BP}(\mathbf{e}, \mathbf{e}') = \min\left(\exp\left(1 - \frac{|\,\mathbf{e}'\,|}{|\,\mathbf{e}\,|} \right), 1 \right) \tag{1.6.3}$$

where $\mathcal{W}_n(\mathbf{e})$ is the set of n-grams of size n in $\mathbf{e}$, $\#_{\mathbf{w}}(\mathbf{e})$ represents the count of n-gram $\mathbf{w}$ in translation $\mathbf{e}$, and $|\,\mathbf{e}\,|$ denotes its length.

In practice, BLEU is defined over complete documents rather than individual sentences. First, n-gram counts are summed up for all sentences in the document. Then, a BLEU score is computed according to Equation (1.6.1) using these document-level counts.

Translation Edit Rate (TER)

TER [Snover et al., 2006] measures translation quality as the amount of editing that is needed to change the candidate translation so it exactly matches the reference translation. Possible edits include the insertion, deletion, and substitution of single words as well as shifts of word sequences. A shift moves a contiguous sequence of words to another location within the hypothesis. All edits, including shifts of any number of words or by any distance, have equal cost. To obtain a dimensionless quantity, the required number of edit operations is usually divided by the total number of words in the reference translation.

[d] An n-gram is a sequence of n consecutive words in a sentence.

[e] Papineni et al. [2002] obtained the best correlation with human judgements using n-grams of maximum size $n = 4$.

In contrast to BLEU, TER is an error score where a 0% value denotes a perfect matching between the candidate translation and the reference. Additionally, TER has a more direct interpretation for people outside of the MT community: the amount of work needed to correct the translations.

1.6.2 Supervision Effort

In addition to the quality of the generated translations, CAT systems have to take into account the cognitive effort required from the user during the translation process. Given that a direct measurement of the cognitive effort cannot be done, supervision effort evaluation is still an open problem. A direct measure that can be used to estimate cognitive effort is the actual time it takes the user to amend the system's outputs. However, since this measure requires the intervention of a human expert, it is slow and expensive which hinders its broad application for research or system development.

Instead, as for translation quality evaluation, MT supervision effort is usually estimated by comparing candidate translations with reference translations. In this case, reference translations can be interpreted as the translations the user may want to obtain. Examples of those measures are the *word-stroke ratio* (WSR) [Tomás and Casacuberta, 2006] and the *key-stroke and mouse-action ratio* (KSMR) [Barrachina et al., 2009].

Both these measures estimate supervision effort as the number of actions performed by a simulated user to obtain the desired translation. The so-called user model defines the possible actions that can be performed by the simulated user. Assuming an IMT interaction protocol (see Figure 1.3), these actions include the detection of errors in the suggested translation (and moving to that position), and the correction of those errors (typing). Conceptually, the former action accounts for the cognitive part of the supervision process while the latter accounts for the actual physical effort required to introduce the corrections. In practice, the search for an error is simulated by computing the longest common character prefix between the translation suggested by the MT system and the reference translation. Then, the first mismatch (word or character) is replaced by the corresponding reference sequence. Finally, the MT system takes into account the validated prefix and the replaced subsequence to suggest a new suffix. This process is iterated until a full match with the reference translation is obtained. Each computation of the longest common prefix would correspond to the user looking for the next error and moving the pointer to the corresponding position of the translation hypothesis. Each sequence replacement, on the other hand, would correspond to a correction typed by the user.

Tomás and Casacuberta [2006] when defining WSR consider that effort required to identify an error is negligible in comparison with the effort required to correct it. Hence, WSR is defined as the quotient between the number of words a user would need to type (word-strokes) in order to obtain the correct translation, and the total number of words in that final correct translation. In this context, a word-stroke is interpreted as a single action and is assumed to have constant cost independently of the length of the typed word.

In contrast, KSMR is calculated as the number of typed characters (key-strokes) plus the number of movements (mouse-actions) divided by the total number of characters of the final translation. This is a more fine-grained measure than WSR since it takes into account both the complexity of the user corrections (number of key-strokes) and the number of them (number of mouse actions) performed to amend the candidate translation. Alternatively, we can ignore the number of mouse-actions and compute simply the number of key-strokes divided by the total number of characters. This measure is known as *key-stroke ratio* (KSR). As a simplification, KSMR assumes that both the search for an error and each key-stroke have equal cost. From the user point of view the two types of actions are different, and may require different types of effort [Macklovitch, 2006]. A weighted measure could take this into account; however, in our experiments, we follow the standard implementation described in [Barrachina et al., 2009] and assume that each action has unit cost.

In addition to the pure effort evaluation of an IMT systems, it is also interesting in some scenarios to estimate the potential supervision effort reduction with respect to a conventional decoupled post-edition CAT system. For this purpose, the post-edition equivalents to WSR and KSR are the *word error rate* (WER) and the *character error rate* (CER) respectively. Both WER and CER are defined as the Levenshtein distance [Levenshtein, 1966] between the candidate translation and a reference translation; WER measures the distance at the word level while CER operates at the character level.

However, CER constitutes a rough estimation of the post-edition effort, since professional translators typically use text editors with autocompletion capabilities to generate the target translations. This problem can be solved by computing the *post-editing key stroke ratio* (PKSR) [Romero et al., 2010]. Here, when the user of the post-edition system enters a character to correct some incorrect word, the system automatically completes the word with the most probable word in the task vocabulary. PKSR is then computed as the number of key-strokes that the user of such post-editing system with word-autocompletion must enter to achieve the reference translation, divided by the total number of reference characters. From this definition, we see that PKSR

and KSR are fairly comparable and the relative difference between them gives us a good estimate of the reduction in human effort that can be achieved by using IMT instead of a conventional decoupled post-edition system.

1.6.3 Statistical Significance of Results

In order to establish that an observed performance difference between two methods is significant, and has not just arisen by chance, we need to apply statistical significance testing. The usual methodology is to state as null hypothesis something like: "The output of methods A and B do not differ with respect to the evaluation measure of interest". Then, we determine the probability, namely the p-value, that the observed difference in the evaluation metric has arisen by chance given the null hypothesis. If the p-value is lower than a predefined significance level (usually $p < 0.05$, or $p < 0.01$) we can reject the null hypothesis.

We use randomization tests for significance testing [Noreen, 1989]. Randomization tests are a class of computer-intensive statistical methods which can compute p-values for complex evaluation measures such as BLEU where analytical methods fail. Randomization test automatically generate sample distributions by randomly shuffling observed data points between experimental conditions. For small test sets, one can enumerate all possible shuffles and compute an exact randomization. For many practical purposes, this is not possible and we must resort to approximate randomization where the collection of test statistics is based on a large enough number of shuffles.

We use an (approximate) randomization version of the paired t-test. Algorithm 1.1 depicts the implementation of this randomization test based on [Chinchor, 1992]. Initially, we use an evaluation measure $Q(\cdot)$ (e.g. BLEU) to determine the absolute difference between the original outcomes of methods A and B. Then, we repeatedly create shuffled versions A' and B' of the original outcomes, determine the absolute difference between their evaluation metrics, and count the number of times N' that this difference is equal or larger than the original difference. In our experiments, the number of repetitions N is equal to $10,000$. To create the shuffled versions of the data sets, we iterate over each data point in the original outcomes and decide based on a simulated coin-flip whether data points should be exchanged between A and B. Finally, the p-value is the proportion of iterations in which the absolute difference in evaluation metric was indeed larger for the shuffled version (corrected to achieve an unbiased estimate).

Algorithm 1.1: Pseudo-code of the randomized paired-samples t-test.

input : A, B (output of the methods under comparison)
$Q(\cdot)$ (evaluation measure function)
N (number of repetitions)
output : p-value of the observed performance difference
auxiliary : Shuffle(A, B) (returns a shuffled version of A and B where some data points have been exchanged)

begin
  $\Delta \leftarrow |Q(A) - Q(B)|$;
  $N' \leftarrow 0$;
  for $1 \leq n \leq N$ **do**
    $A', B' \leftarrow$ Shuffle(A, B);
    $\Delta' \leftarrow |Q(A') - Q(B')|$;
    if $\Delta' \geq \Delta$ **then**
      $N' \leftarrow N' + 1$;
  return $\frac{N'+1}{N+1}$; `// 1 is added to achieve an unbiased estimate`

1.7 Scientific Goals

We have seen that despite the intensive research effort invested in the last fifty years, MT technology is still error-prone. This thesis is devoted to study different approaches and strategies to improve the broader and more efficient deployment of currently imperfect MT technology.

Initially, we focus on the improvement of fully-automatic MT technology. Particularly, we study the combination of multiple MT systems to generate translations of higher quality. The key idea of system combination [Dietterich, 2000] is that it is often very difficult to find the real best system for the task at hand, while different systems (for instance, trained on different data or using different learning paradigms) can exhibit complementary strengths and limitations. Therefore, a proper combination of various systems could be more effective than using a single monolithic system.

Then, we focus on improving the utility of automatic translations for the end-user. To do that, we study current QE technology and identify several potential problems due to the features employed to perform the prediction. We propose a two-step training procedure designed to deal systematically with the usually highly-redundant sets of features that hinder the learning process of QE models. The keystone of this training methodology is the use of a dimensionality reduction (DR) module to obtain a set of features that allows for a robust training of the QE system. Hence, we also investigate on DR tech-

niques that allows to efficiently extract, from a set of ambiguous and redundant features, the latent variables that actually govern translation quality.

Finally, we investigate techniques to improve the productivity of CAT systems. We focus on IMT technology, and particularly, on the development of alternative interaction protocols intended to improve the overall translation performance. Instead of the conventional *passive* interaction depicted in Figure 1.3 where the human expert was assumed to exhaustively supervise[f] each system suggestion, we studied two different *active* protocols where the system decides for which hypothesis may be worth to ask for user supervision.

1.7.1 Combination of Machine Translation Systems

As we have shown in Section 1.2, many different MT approaches have been proposed in the literature [Hutchins and Somers, 1992; Somers, 2003; Lopez, 2008]. Rule-based approaches are expensive to develop and are usually too rigid to translate sentences from a general domain. However, they are particularly effective in dealing with semantic, morphological, and syntactic phenomena. In contrast, SMT systems are more robust in processing partial and/or ill-formed sentences, but they use no linguistic background and have difficulties in capturing long distance phenomena. Example-based systems heavily depend on the quality of collected examples and the similarity measures between examples and input sentences. However, they can be very accurate on inputs that match an example. From the viewpoint of MT system designers, if we could integrate the advantages of these approaches and get rid of their disadvantages, that combined system could perform better than any of the individual systems.

The idea of system combination, known as ensemble learning [Opitz and Maclin, 1999] by the machine learning community, has been investigated in the pattern recognition field since the late seventies when Tukey [1977] suggests combining two linear regression models. Since then, system combination have been shown to be quite a successful pattern recognition technique [Roth and Zelenko, 1998; Larkey and Croft, 1996; Fiscus, 1997]. However, several problems arise when combining the structured outputs of MT systems.

The combination of structured outputs involves two main challenges: the detection of the "best" parts of the provided outputs, and the combination of these parts to generate the final consensus output. Since MT outputs, namely target language sentences, may have different lengths and different word orders, an alignment is additionally needed to synchronize them. Then, an ap-

[f] We use the term *supervision* to denote the interactive translation process of IMT systems.

propriate decision function has to be implemented to obtain the consensus translation from the result of the synchronization [Bangalore, 2001; Jayaraman and Lavie, 2005; Rosti et al., 2007a; Sim et al., 2007; Matusov et al., 2008]. We will use the term *subsequence-combination* to denote this type of combination methods.

Subsequence-combination systems must address very challenging problems, particularly the above-mentioned alignment step. Therefore, some system combination methods for MT [Callison-burch and Flournoy, 2001; Nomoto, 2004; Paul et al., 2005] ignore the synchronization step and simply select one of the provided translations. In exchange, these latter methods can implement sophisticate classifiers (such as minimum Bayes' risk classifiers) which constitutes their main virtue. We will refer to these approaches as *sentence-selection* methods.

We propose a new system combination method that gathers together the sophisticated search algorithms of sentence-selection methods and the ability to generate new improved translations of subsequence-combination methods. Our method combines several MT systems by detecting the "best" parts of the systems' translations and combining them into a (possibly new) consensus translation which is optimal with respect to a particular performance measure. Chapter 2 is devoted to describe and evaluate the proposed system combination method.

1.7.2 Machine Translation Quality Estimation

Quality estimation is typically addressed as a regression problem [Quirk, 2004; Blatz et al., 2004; Specia et al., 2009b]. Given a translation generated by an MT system (and potentially other additional sources of information) a set of features is extracted. Then, a model trained using a particular machine learning algorithm is employed to compute a quality score from these features. Most works on QE consider a fixed set of features and study the performance of different learning algorithms on those features. However, feature sets tend to be highly redundant, i.e. there is high multicollinearity between the features, and some of the features may even be irrelevant to predict the quality score. Moreover, a set of translations labeled with their *True* quality score is required to train the learning model. Since this labeling process is usually done manually, training sets rarely contain enough labeled samples to accurately train the model. By removing irrelevant and redundant features from the data, DR methods potentially improve the performance of learning models by alleviating the effect of the so-called "curse" of dimensionality [Bellman, 1961], enhancing the generalization capability of the model, and speeding up

the learning process. Additionally, DR may also help the researchers to acquire better understanding about their data by telling them which are the important features and how they are related with each other. Despite these potential improvements, works on QE usually put little attention on DR techniques.

In Chapter 3, we start by proposing a two-step training methodology whose goal is to address the learning problems inherent to several natural language tasks. Then, we propose two novel DR methods based on partial least squares regression (PLSR) [Wold, 1966] that will act as cornerstones of the proposed training methodology. Finally, we apply the proposed methodology to study the performance of the proposed DR methods in a translation QE task. Additionally, we also study how the use of DR methods affect the performance of different learning models.

1.7.3 Active Protocols for Interactive Machine Translation

Despite being an efficient CAT implementation, conventional IMT technology still requires the human expert to systematically supervise each successive hypothesis in order to find the point where the next translation error appears. From the system's point of view this interaction protocol is considered *passive* because the system just waits for the human feedback without concern about how supervision decisions are taken. Clearly, with a passive protocol, perfect results from the human point of view can be guaranteed, because it is the user who is fully responsible of the accurateness of these results. However, we must take into account that each translation supervision involves the user reading and understanding the proposed target language sentence and deciding if it is an adequate translation of the source sentence, which, even in the case of error-free translations, is a process that requires a non-negligible cognitive effort.

As an alternative, we study the implementation of *active* protocols into IMT systems. In an active protocol, the system is able to proactively inform the user about which translation elements (full translations or subsequences of them) should be supervised. In contrast to passive interaction, the translations generated using an active protocol may be different from the ones the user has in mind. However, an adequate selection of translation elements may provide better compromises between overall human effort and final translation quality, hence, optimizing the overall system-human performance. This is one of the main potential advantages of active protocols since it allows us to adapt the system according to the requirements of a given task and/or level of expertise of the user.

Chapter 4 is devoted to describe an active interaction protocol where the

system informs the user about the reliability of the suggested translations. This reliability estimation is then considered as an additional source of information to guide the user in her interaction with the system. In Chapter 5 we further explore these idea to develop an active learning framework for IMT with the objective of generating translations of the highest quality at the lowest human effort possible. The proposed active learning framework estimates the utility of supervising each automatic translation so that we optimize the translation accuracy of a dynamic SMT model updated with each available user supervised translation.

1.8 Summary

In this chapter we have introduced the field of MT classifying the main MT approaches that have been proposed so far according to different criteria. We have paid special attention to the statistical approach to MT since it is the approach in which this thesis is focused. We have also introduced the task of QE for MT and its several practical uses. Then, we have motivated the interactive MT approach as an alternative to the fully-automatic SMT approach. Next, we have presented the evaluation measures typically employed to test SMT and IMT systems. Finally, we have described the different research lines explored in this thesis and the specific goals of each of them.

Chapter

2 Minimum Bayes' Risk System Combination

System combination has proved to be a successful technique in the pattern recognition field. However, several difficulties arise when combining the outputs of tasks, e.g. MT, that generate structured patterns. So far, MT system combination approaches either implement sophisticated classifiers to select one of the provided translations, or generate new sentences by combining the best subsequences of the provided translations. We present *minimum Bayes' risk system combination* (MBRSC), a system combination method for MT that gathers together the advantages of sentence-selection and subsequence-combination methods. MBRSC is able to detect and utilize the best subsequences of the provided translations to generate the optimal consensus translation with respect to a particular performance metric.

This chapter is devoted to discuss the modeling and implementation problems encountered wile developing MBRSC. Section 2.1 introduces system combination for MT, Section 2.2 presents the MBRSC model and its MBR formulation for BLEU. Section 2.3 studies the computational complexity of the BLEU-based risk computation and provides different approaches to efficiently obtain it. Section 2.4 describes the search problem and proposes several algorithms to obtain the optimal consensus translation. Section 2.5 presents the results of the experimentation carried out to test the proposed system combination approach. Finally, we provide a summary of the chapter in Section 2.6.

Chapter Outline

2.1 Introduction

Despite a major development boost in the early nineties, state-of-the-art MT systems are still far from perfect [NIST, 2006; Callison-Burch et al., 2008, 2012]. The combination of multiple MT systems is a promising research direction to improve the accuracy of current MT technology. The key idea of system combination [Dietterich, 2000] is that it is often very difficult to find the real best system for the task at hand, while different systems (for instance, trained on different data or using different learning paradigms) can exhibit complementary strengths and limitations. Therefore, a proper combination of various systems could be more effective than using a single system.

The combination of outputs from multiple systems have been found to improve accuracy in a number of classification task such as part-of-speech tagging [Roth and Zelenko, 1998], text categorization [Larkey and Croft, 1996] and speech recognition [Fiscus, 1997]. However, unlike part-of-speech tagging or text categorization where the classes are atomic units (either a part-of-speech or a category), classes in a translation task are sequences (sentences of words). When combining MT systems, we can consider either the full sentence or the individual words as the atomic classes, which leads to two different system combination approaches.

MT system combination methods that consider the full sentences as the classification classes implement the so-called sentence selection approach. The decision on the consensus translation is taken as a selection of a translation provided by one of the individual MT systems [Callison-burch and Flournoy, 2001; Nomoto, 2004; Paul et al., 2005; DeNero et al., 2010; Duan et al., 2010]. Their main limitation is that they cannot generate new translations that include "good" subsequences from different individual sentences. In exchange, they can implement sophisticated classifiers such as minimum Bayes' risk classifiers [Duda et al., 2001], which constitutes their main virtue.

In contrast, MT system combination methods that consider the individual words as the classification classes implement the so-called subsequence-combination approach. These methods are able to detect which words (or sequences thereof) of the individual translations are correct, and combine these subsequences to generate a consensus translation with reduced error [Fiscus, 1997]. Unfortunately, the translations provided by the individual systems can be of different length or have a different word order. Therefore, a synchronization (alignment) step is required to detect which is the correspondence between the subsequences of the different translations. The consensus translation is given by the highest scoring path through the graph, the so-called

confusion network, defined by the computed alignment [Bangalore, 2001; Jayaraman and Lavie, 2005; Rosti et al., 2007b; Matusov et al., 2008; He et al., 2008]. Subsequence-combination methods have one obvious advantage over sentence-selection: they can generate new consensus translations that contain the best subsequences of the individual translations. However, they have to deal with the challenging alignment problem that, since the consensus translation is bounded to the structural restrictions of the alignment, has a substantial effect on combination performance [He et al., 2008]. Moreover, these methods also require additional data to train complex search models that score the paths thought the consensus network, which hinders their application to languages with scarce resources.

We present *minimum Bayes' risk system combination* (MBRSC), a method to generate consensus translations designed to gather together the advantages of sentence-selection and subsequence-combination methods. MBRSC can detect the best subsequences of the provided translations, and combine them into a new consensus translation which is optimal with respect to a particular performance measure. We choose the BLEU score [Papineni et al., 2002] as our performance measure of interest. BLEU considers a sentence as a vector of n-gram occurrences rather than a word sequence. In other words, BLEU can compare sentences knowing only their n-grams and it does not require an explicit alignment between the sentences. Additionally, BLEU is the standard performance measure for MT, thus, by using it as our loss function, we are optimizing our system towards the most widespread translation quality measure.

In comparison with sentence-selection methods, MBSRC also implements a sophisticated classifier, and, additionally, it is able to generate new consensus translations that include the best subsequences from different individual translations. Regarding subsequence-combination methods, MBRSC has several advantages over the dominant confusion network approach:

- Translations do not have to be synchronized which avoids the limitations imposed by the alignment.
- The full target language is explored in the search for the consensus translation.
- A minimum Bayes' risk classifier is implemented. Thus, the consensus translations are optimal with respect to the final evaluation measure.
- No additional data is required to train graph-search models.

Following sections describe the formalization of MBRSC as a minimum Bayes' risk classifier for an ensemble of MT systems using BLEU as loss function. Particularly, how this BLEU-based risk is implemented, and the algorithms used to efficiently explore the target language to obtain the MBR translation.

2.2 MBRSC Model

Let $\{\mathrm{C}_1, \ldots, \mathrm{C}_k, \ldots, \mathrm{C}_K\}$ denote K individual MT systems. Under the assumption that the systems are statistically independent, we model the multi-system classifier as a weighted ensemble of probability distributions [Kittler et al., 1998]:

$$\mathrm{P}(\mathbf{e} \mid \mathbf{f}) = \sum_{k=1}^{K} \alpha_k \cdot \mathrm{P}_k(\mathbf{e} \mid \mathbf{f}) \tag{2.2.1}$$

where $\mathbf{f}$ represents a sentence in the source language $\mathcal{F}$, $\mathbf{e}$ represents a translation in the target language $\mathcal{E}$, and $\mathrm{P}_k(\mathbf{e} \mid \mathbf{f})$ denotes the probability distribution over translations modeled by system C_k. The free parameters of the ensemble model $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_K\}$ are scaling factors that can be interpreted as a measure of the importance of each individual system ($\sum_{k=1}^{K} \alpha_k = 1$).

Given the ensemble model in Equation (2.2.1) and a loss function $\mathrm{L}(\mathbf{e}, \mathbf{e}')$, the corresponding optimal classification function is an instance of the MBR classifier in Equation (1.3.4):

$$\begin{aligned}
\hat{\mathbf{e}} &= \arg\min_{\mathbf{e} \in \mathcal{E}} \mathrm{R}(\mathbf{e} \mid \mathbf{f}) \\
&\approx \arg\min_{\mathbf{e} \in \mathcal{E}} \sum_{\mathbf{e}' \in \mathcal{E}} \left(\sum_{k=1}^{K} \alpha_k \cdot \mathrm{P}_k(\mathbf{e}' \mid \mathbf{f}) \right) \cdot \mathrm{L}(\mathbf{e}, \mathbf{e}') \\
&= \arg\min_{\mathbf{e} \in \mathcal{E}} \sum_{k=1}^{K} \alpha_k \cdot \underbrace{\left(\sum_{\mathbf{e}' \in \mathcal{E}} \mathrm{P}_k(\mathbf{e}' \mid \mathbf{f}) \cdot \mathrm{L}(\mathbf{e}, \mathbf{e}') \right)}_{\text{system-specific loss}}
\end{aligned} \tag{2.2.2}$$

where the factor between parenthesis denotes the expected loss of candidate translation $\mathbf{e}$ according to system k. Note that Equation (2.2.1) and Equation (2.2.2) assume that all individual systems share the same space of translation ($\mathcal{E}$) which in practice is always not true. For the sake of simplicity, we will skip over this practical problem for now; Section 2.3.3 will describe our approach to deal with this challenge.

The common approach would be to particularize Equation (2.2.2) to use the $0-1$ loss function. However, while the use of the $0-1$ loss function aims at a minimization of the sentence error rate (see Section 1.3), most MT systems are evaluated by their ability to minimize the error rate at word level (TER [Snover et al., 2006]) or n-gram level (BLEU [Papineni et al., 2002]). Therefore, instead of the $0-1$ function, we choose the widespread BLEU score as the loss function of interest. Note that sentence-level BLEU is a gain function that returns a percentage with a value of one denoting an exact match between $\mathbf{e}$ and $\mathbf{e}'$ (see Section 1.6.1). Thus, we have to substitute the $\arg\min_{\mathbf{e}\in\mathcal{E}}$ operator in Equation (2.2.2) by an $\arg\max_{\mathbf{e}\in\mathcal{E}}$ operator. The BLEU-based MBR classifier for the ensemble is finally formulated as:

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}\in\mathcal{E}} \mathrm{R}(\mathbf{e} \mid \mathbf{f}) \\ &\approx \arg\max_{\mathbf{e}\in\mathcal{E}} \sum_{k=1}^{K} \alpha_k \cdot \left(\sum_{\mathbf{e}'\in\mathcal{E}} \mathrm{P}_k(\mathbf{e}' \mid \mathbf{f}) \cdot \mathrm{BLEU}(\mathbf{e}, \mathbf{e}') \right) \end{aligned} \tag{2.2.3}$$

We use the minimum error rate training (MERT) [Och, 2003] criterion to optimize the free parameters $\boldsymbol{\alpha}$ of the MBRSC model in Equation (2.2.1). Our goal is to obtain the values of the parameters $\boldsymbol{\alpha}$ that maximize the quality of the consensus translations generated by MBRSC for a representative training set $\{(\mathbf{f}_n, \mathbf{e}_n)\}_{n=1}^{N}$:

$$\widehat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \sum_{n=1}^{N} \mathrm{BLEU}(\hat{\mathbf{e}}_{\boldsymbol{\alpha}}, \mathbf{e}_n) \tag{2.2.4}$$

where $\hat{\mathbf{e}}_{\boldsymbol{\alpha}}$ denotes the consensus translation for source sentence $\mathbf{f}_n$ given by the MBRSC decision function (Equation (2.2.3)) using parameter values $\boldsymbol{\alpha}$. Note that we use BLEU as quality function, but any other function can be used as well. Finally, we solve this optimization problem with the downhill-simplex algorithm [Nelder and Mead, 1965].

The MBR classifier in Equation (2.2.3) has a very high temporal complexity in $O(|\,\mathcal{E}\,|^2 \cdot I)$, where $|\,\mathcal{E}\,|$ denotes the number of possible target language sentences, and I represents the maximum sentence length given that $\mathrm{BLEU}(\mathbf{e}, \mathbf{e}')$ can be computed in $O(\max(|\,\mathbf{e}\,|, |\,\mathbf{e}'\,|))$ time. The most troublesome factor in this complexity is given by the squared $O(|\,\mathcal{E}\,|^2)$. Since the number of sentences in the target language is potentially infinite, an exhaustive enumeration of all these sentences is unfeasible. Thus, we must define efficient methods to manage this potentially infinite space of translations. On the one hand, Section 2.3 present several techniques developed to efficiently compute the BLEU-based risk for a given candidate translation. On the other

hand, Section 2.4 describe various search algorithms whose goal is to obtain the optimal consensus translation.

2.3 MBRSC Risk Computation

As explained in Section 1.3.1, given a loss function $\mathrm{L}(\mathbf{e}, \mathbf{e}')$ the optimal classification function is the one that minimizes the Bayes' risk for each object [Bickel and Doksum, 1977], in our case:

$$\mathrm{R}(\mathbf{e} \mid \mathbf{f}) = \sum_{\mathbf{e}' \in \mathcal{E}} \Pr(\mathbf{e}' \mid \mathbf{f}) \cdot \mathrm{L}(\mathbf{e}, \mathbf{e}') \tag{2.3.1}$$

This risk computation for general loss functions is extremely costly for MT, where the number of classes $|\,\mathcal{E}\,|$ is potentially infinite. This is the reason why the vast majority of SMT systems prefer to use the $0-1$ loss function despite being particularly inadequate to evaluate MT outputs. However, the complexity of Equation (2.3.1) can be greatly reduced if instead of general loss functions we consider linear loss functions over sentence features. Given a loss function of the form $\mathrm{L}(\mathbf{e}, \mathbf{e}') = \sum_m \theta_m(\mathbf{e}) \cdot \phi_m(\mathbf{e}')$, where $\phi_m(\mathbf{e}')$ are real-valued features of the reference translation $\mathbf{e}'$, and $\theta_m(\mathbf{e})$ are sentence-specific weights on those features of the candidate translation $\mathbf{e}$, the risk computation in Equation (2.3.1) can be re-written as:

$$\begin{aligned}
\mathrm{R}(\mathbf{e} \mid \mathbf{f}) &= \sum_{\mathbf{e}' \in \mathcal{E}} \Pr(\mathbf{e}' \mid \mathbf{f}) \cdot \sum_m \theta_m(\mathbf{e}) \cdot \phi_m(\mathbf{e}') \\
&= \sum_m \theta_m(\mathbf{e}) \cdot \sum_{\mathbf{e}' \in \mathcal{E}} \Pr(\mathbf{e}' \mid \mathbf{f}) \cdot \phi_m(\mathbf{e}') \\
&= \sum_m \theta_m(\mathbf{e}) \cdot \mathbb{E}_{\Pr(\mathbf{e}'|\mathbf{f})}[\phi_m(\mathbf{e}')]
\end{aligned} \tag{2.3.2}$$

Since the feature expectations $\mathbb{E}_{\Pr(\mathbf{e}'|\mathbf{f})}[\phi_m(\mathbf{e}')]$ can be precomputed in advance, the risk computation in Equation (2.3.2) implies that we can find MBR translations by first computing all feature expectations and then computing the risk of each candidate translation in a single function computation. The time complexity of such search process is $O(|\,\mathcal{E}\,| \cdot I)$ assuming that the number of non-zero features ϕ_m and hypothesis-specific counts θ_m grow linearly in sentence length I and all of them can be computed in constant time. This is an important reduction respect to the complexity, $O(|\,\mathcal{E}\,|^2 \cdot I)$, of an MBR classifier that computes the risk according to Equation (2.3.1).

Unfortunately, many loss functions of interest (e.g. BLEU) are not linear, and so Equation (2.3.2) does not apply. However, they usually are functions

of some features of $\mathbf{e}'$. For example, BLEU in Equations (1.6.2) and (1.6.3) references $\mathbf{e}'$ only via its n-gram counts $\#_{\mathbf{w}}(\mathbf{e}')$. Therefore, these loss functions can be expressed as $\widetilde{\mathrm{L}}(\mathbf{e}, \Phi(\mathbf{e}'))$ for a feature mapping $\Phi : \mathcal{E} \rightarrow \mathbb{R}^m$. Based on this observation, DeNero et al. [2009] proposed to follow the structure of Equation (2.3.2) also for such nonlinear evaluation functions, computing the risk of $\mathbf{e}$ based on the corresponding feature expectations of $\mathbf{e}'$:

$$\mathrm{R}(\mathbf{e} \mid \mathbf{f}) \approx \widetilde{\mathrm{L}}(\mathbf{e}, \mathbb{E}_{\mathrm{Pr}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')]) \tag{2.3.3}$$

Note that for nonlinear loss functions, this risk computation over features differs from the exact risk in Equation (2.3.1), but MT decoding results reported in [DeNero et al., 2009] showed that there were no significant difference in performance between the two approaches.

We explore this risk formulation in two directions to deal with the nonlinear BLEU function. First, we implement a first-order vector Taylor series expansion to approximate the BLEU score by a linear function (Section 2.3.1). Under this approach, we can exactly compute the risk of the candidate translations, but in exchange we have to use a function that differs from the final evaluation score. Second, we apply the risk computation over features directly for BLEU (Section 2.3.2). In this case, we only can compute an approximation to the true risk value, but this approximation is computed using the exact final evaluation measure.

2.3.1 Linear BLEU

The risk computation over features relies on the use of a linear loss function. Here, we describe a linear approximation to the logarithm of the BLEU score as done in [Tromble et al., 2008]. The authors start by defining the following linear gain function:

$$\mathrm{G}(\mathbf{e}, \mathbf{e}') = \lambda_0 \cdot |\,\mathbf{e}\,| + \sum_{n=1}^{4} \sum_{\mathbf{w} \in \mathcal{W}_n} \lambda_{\mathbf{w}} \cdot \#_{\mathbf{w}}(\mathbf{e}) \cdot \delta_{\mathbf{w}}(\mathbf{e}') \tag{2.3.4}$$

where $\lambda_{\mathbf{w}}$ are constants, and $\delta_{\mathbf{w}}(\mathbf{e}')$ is an indicator feature whose value is equal to one if $\mathbf{w}$ is present in $\mathbf{e}'$ and zero otherwise. The indicator features $\delta_{\mathbf{w}}(\mathbf{e}')$ are the real-valued features of the reference class (ϕ_m) while the n-gram counts $\#_{\mathbf{w}}(\mathbf{e})$ are the candidate-specific counts on those features (θ_m). Using this gain function in place of the loss function in Equation (2.3.1), the risk for a candidate translation $\mathbf{e}$ can be rewritten as follows:

$$\mathrm{R}(\mathbf{e} \mid \mathbf{f}) = \lambda_0 \cdot |\,\mathbf{e}\,| + \sum_{n=1}^{4} \sum_{\mathbf{w} \in \mathcal{W}_n} \lambda_{\mathbf{w}} \cdot \#_{\mathbf{w}}(\mathbf{e}) \cdot \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')] \tag{2.3.5}$$

where $\mathrm{P}(\mathbf{e}' \mid \mathbf{f})$ (Equation (2.2.1)) denotes the probability distribution of the ensemble, λ_0 and $\lambda_{\mathbf{w}}$ are model parameters, and $\mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')]$ denotes the expected probability of n-gram $\mathbf{w}$ to be present according to $\mathrm{P}(\mathbf{e}' \mid \mathbf{f})$. Note that similarly to Equation (2.2.3), the risk in Equation (2.3.5) denotes the expected translation quality, as measured by linear BLEU, of the candidate translation $\mathbf{e}$.

The value of free parameters $\lambda_0, \lambda_{\mathbf{w}}$ can be computed from the 1-gram precision p, the ratio in which the n-gram precisions exponentially decay r, and the number of 1-gram tokens T. A detailed explanation of how these values are obtained is given in Appendix B.

$$\lambda_0 = \frac{-1}{T} \qquad \lambda_{\mathbf{w}} = \frac{1}{4 \cdot T \cdot p \cdot r^{|\mathbf{w}|-1}} \tag{2.3.6}$$

To avoid the dependence on a particular decoding run, values p and r are usually averaged across multiple development sets. Substituting the above factors in Equation (2.3.5), we obtain the following risk formulation for the linear BLEU gain function in Equation (2.3.4):

$$\mathrm{R}(\mathbf{e} \mid \mathbf{f}) = -\frac{|\mathbf{e}|}{T} + \sum_{n=1}^{4} \sum_{\mathbf{w} \in \mathcal{W}_n} \frac{\#_{\mathbf{w}}(\mathbf{e}) \cdot \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')]}{4 \cdot T \cdot p \cdot r^{|\mathbf{w}|-1}} \tag{2.3.7}$$

This equation computes the exact risk of a candidate translation $\mathbf{e}$. Additionally, it can be computed incrementally which will simplify the implementation of some search algorithms that will be described in Section 2.4. In contrast, the main drawback of this formulation is that it uses a gain function that is an approximation to the BLEU score. Equation (2.3.4) ignores the count clipping present in the exact BLEU score where a correct n-gram present once in the reference but several times in the candidate translation will be counted only once as correct.

2.3.2 BLEU over n-gram Count Expectations

Linear BLEU allows us to efficiently compute the exact risk for a function that, due to the lack of n-gram counts clippings in its formulation, approximates the final evaluation measure BLEU. In contrast, DeNero et al. [2009] proposed an alternative approach (see Equation (2.3.3)) where the exact risk in Equation (2.2.3) is approximated using the exact BLEU formulation.

BLEU (see Equation (1.6.1)) utilizes the reference translation $\mathbf{e}'$ only via its n-gram counts[a] so it can be expressed as $\widetilde{\mathrm{BLEU}}(\mathbf{e}, \Phi(\mathbf{e}'))$ for a feature map-

[a]The brevity penalty is also a function of n-gram counts: $|\mathbf{e}'| = \sum_{\mathbf{w} \in \mathcal{W}_1(\mathbf{e}')} \#_{\mathbf{w}}(\mathbf{e}')$.

ping $\Phi : \mathcal{E} \to \mathbb{R}^m$ where each sentence is mapped to its set of n-gram counts. Thus, in contrast with linear BLEU where the reference features were indicator functions, see Equation (2.3.7), now the reference features are the counts of the n-grams in each sentence $\Phi(\mathbf{e}') = \{\#_{\mathbf{w}}(\mathbf{e}') \mid 1 \geq n \geq 4 \wedge \mathbf{w} \in \mathcal{W}_n(\mathbf{e}')\}$. Formally, this BLEU-based risk over n-gram count features is given by:

$$\begin{aligned} \mathrm{R}(\mathbf{e} \mid \mathbf{f}) &\approx \widetilde{\mathrm{BLEU}}(\mathbf{e}, \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')]) \\ &= \left(\prod_{n=1}^{4} \widetilde{\mathrm{PR}}_n(\mathbf{e}, \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')])\right)^{\frac{1}{4}} \cdot \widetilde{\mathrm{BP}}(\mathbf{e}, \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')]) \end{aligned} \tag{2.3.8}$$

where $\mathrm{P}(\mathbf{e}' \mid \mathbf{f})$ is the ensemble model in Equation (2.2.1). Consequently, we reformulate the n-gram precisions $\mathrm{PR}_n(\mathbf{e}, \mathbf{e}')$ and the brevity penalty $\mathrm{BP}(\mathbf{e}, \mathbf{e}')$ as functions of expected n-gram counts:

$$\widetilde{\mathrm{PR}}_n(\mathbf{e}, \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')]) = \frac{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{e}')} \min(\#_{\mathbf{w}}(\mathbf{e}), \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\#_{\mathbf{w}}(\mathbf{e}')])}{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{e}')} \#_{\mathbf{w}}(\mathbf{e})} \tag{2.3.9}$$

$$\widetilde{\mathrm{BP}}(\mathbf{e}, \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\Phi(\mathbf{e}')]) = \min\left(\exp\left(1 - \frac{\mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[|\,\mathbf{e}'\,|]}{|\,\mathbf{e}\,|}\right), 1\right) \tag{2.3.10}$$

The main advantage of the BLEU-based risk formulation in Equation (2.3.8) is that it uses the actual BLEU score to compute the risk. For example, it takes into account the count clipping ($\min(\cdot)$ functions) present in the BLEU formulation (Equations (1.6.2) and (1.6.3)). Thus, we score the candidate translations with the same measure that will be used to evaluate the system. In exchange, since BLEU is a nonlinear function, the formulation in Equation (2.3.8) differs from the exact BLEU-based risk score in Equation (2.3.1). Also, Equation (2.3.8) can not be computed incrementally which will add an extra complexity factor to some search algorithms.

2.3.3 Computing Feature Expectations

We have described two different approaches (Equations (2.3.7) and (2.3.8)) to efficiently compute the BLEU-based risk for a candidate translation $\mathbf{e}'$. The risk computation of both approaches is based on the expected values of some n-gram features $\phi_{\mathbf{w}}(\mathbf{e})$: indicator features $\delta_{\mathbf{w}}(\mathbf{e})$ in Section 2.3.1, and count features $\#_{\mathbf{w}}(\mathbf{e})$ in Section 2.3.2. The expected value of these n-gram features

according to the ensemble model is given by:

$$\begin{aligned}\mathbb{E}_{\mathrm{P}(\mathbf{e}|\mathbf{f})}[\phi_{\mathbf{w}}(\mathbf{e})] &= \sum_{\mathbf{e}\in\mathcal{E}} \mathrm{P}(\mathbf{e} \mid \mathbf{f}) \cdot \phi_{\mathbf{w}}(\mathbf{e}) \\ &= \sum_{k=1}^{K} \alpha_k \cdot \sum_{\mathbf{e}\in\mathcal{E}} \mathrm{P}_k(\mathbf{e} \mid \mathbf{f}) \cdot \phi_{\mathbf{w}}(\mathbf{e}) \end{aligned} \tag{2.3.11}$$

According to this equation, the probability distributions $\mathrm{P}_k(\mathbf{e} \mid \mathbf{f})$ of the MT models are considered ideal distributions with the assumption that translation candidates are shared by all systems. However, due to differences in generative capabilities, training data selection, and various pruning techniques, the domain of translations of the different systems are always not identical in practice. We compute the expectations individually for each system and then combine them according to the ensemble weights under the assumption that if a translation $\mathbf{e}$ is not present in the translation domain of system k this implies that $\mathbf{e}$ has zero probability of being generated by system k.

The computation of the expectations for each system depends on the representation selected to express its translation domain. Computing expectations from lists of N-best translations is trivial, the straightforward application of Equation 2.3.11 will do the trick. For more complex representations such as hypergraphs [Huang, 2008], the algorithms proposed in [Kumar et al., 2009; DeNero et al., 2009, 2010] can be used. Appendix C provides a detailed description of these methods.

Finally, if a probability distribution over translations is not available (e.g. it is a rule-based MT system), we can assume a uniform probability distribution or assign a rank-based probability [Rosti et al., 2007b] to each translation.

2.4 MBRSC Search

The goal of the search problem in SMT, also referred to as generation or decoding, is to obtain the optimal target language sentence for a given source sentence according to the chosen translation model. In our case, the translation model is defined by Equation (2.2.3) and the search problem involves to find the translation of maximum expected BLEU score among all possible target language sentences. The main difficulty in the computation of Equation (2.2.3) is the potentially infinite number of target language sentences $\mathbf{e} \in \mathcal{E}$ that have to be considered as candidate translations during the search process. A similar search problem also arises in conventional SMT and has been demonstrated

to be an NP-complete problem [Knight, 1999; Udupa and Maji, 2006], so we cannot expect to develop efficient algorithms to perform an exact search.

The actual search problem (related to the $\arg\max_{\mathbf{e}\in\mathcal{E}}$ operator) can be instantiated to use different risk functions, see Section 2.3. Thus, in the description of the search algorithms, we use the generic symbol $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$ to denote the expected BLEU score, namely the risk, of translation $\mathbf{e}$. The final complexity of the MBRSC search algorithms will depend on the particular formulation chosen to compute $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$, namely exact BLEU risk in Equation (2.2.3), linear BLEU risk in Equation (2.3.7), or BLEU risk over n-gram count expectations in Equation (2.3.8).

The classical MBR approach to deal with the infinite number of candidate translations is to consider only a downsized translation space $\mathcal{C} \subseteq \mathcal{E}$ that contains a finite number of candidate translations [Kumar and Byrne, 2004; Ehling et al., 2007; Tromble et al., 2008; Kumar et al., 2009]. Those works study MBR classifiers for single MT systems, thus $\mathcal{C}$ is a compact encoding, e.g. an N-best list or an hypergraph) of the translation distribution generated by the MT system. In our case, $\mathcal{C}$ is the union of the compact representations provided by the models C_k being combined. Since only a portion of the full translation space $\mathcal{E}$ is explored, the search process is an approximation to its "true" solution, but in exchange, it can be straightforwardly implemented by simple algorithms. This approach is equivalent to the rationale behind sentence selection methods for MT system combination [Callison-burch and Flournoy, 2001; Nomoto, 2004; Paul et al., 2005; DeNero et al., 2010]. Section 2.4.1 describes a search algorithm that follows this approach.

The main drawback of taking into consideration only a finite subset of the translation space is that many low-probability translations are ignored. Since the importance of each translation in computing the risk is proportional to its probability, ignoring low-probability translations do not have a great impact in the computation of the risk. In fact, previous works in MT [Ehling et al., 2007] and automatic speech recognition [Stolcke et al., 1997; Mangu et al., 2000] suggest that the use of a few thousand best candidates is sufficient. However, it may have a great impact in the search for the optimal translation. Moreover, while we aim at obtaining the translation of maximum expected BLEU score, conventional SMT systems search for the translation of maximum probability (see Section 1.3). This mismatch implies that the minimum risk translation does not have to be one of the maximum probability candidate translations in $\mathcal{C}$, and thus performing the search in an extended search space may lead to translations of higher expected BLEU score.

Following this rationale, we can develop methods that use different transla-

Algorithm 2.1: Sentence selection search.

```
input     : f (source language sentence)
            C (finite list of candidate translations for x)
output    : ê, q̂ (best translation along with its score)
auxiliary : R(e | f) (returns the expected BLEU score of translation e)
1 begin
2     q̂ ← −∞;
3     for e ∈ C do
4         q ← R(e | f);
5         if q > q̂ then
6             q̂, ê ← q, e;
7     return ê, q̂;
```

tion spaces for risk computation and search. While a finite translation space is still used to compute the risk, different techniques can be applied to efficiently explore an extended translation space during search. We present search algorithms based on two of these techniques: greedy algorithms [Berger et al., 1994; Germann et al., 2001] in Section 2.4.2, and dynamic programming [Bellman, 1957; Zens et al., 2002] in Section 2.4.3.

2.4.1 Sentence Selection Search Algorithm

The most straightforward approach to reduce search complexity is to represent the potentially infinite search domain $\mathcal{E}$ by a finite, although potentially large, search domain $\mathcal{C}$. Typically $\mathcal{C}$ is a list of candidate translations [Ehling et al., 2007], but more sophisticated representations, such as hypergraphs [Huang, 2008], can also be used. For simplicity, we assume that $\mathcal{C}$ is a list of translations. Then, this search over sentence pairs is depicted in Algorithm 2.1. The running time of Algorithm 2.1 is $O(|\,\mathcal{C}\,| \cdot Z)$, where $O(Z)$ is the computational complexity of $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$.

Following the exact MBR formulation in Equation (2.2.3), the risk is calculated by exhaustively computing the BLEU score between all pairs of sentences in $\mathcal{C}$. The final complexity of the algorithm is then in $O(|\,\mathcal{C}\,|^2 \cdot I)$, where I is the maximum sentence length, given that $\mathrm{BLEU}(\mathbf{e}, \mathbf{e}')$ can be computed in $O(\max(|\,\mathbf{e}\,|, |\,\mathbf{e}'\,|))$ time. If by contrast we follow Equation (2.3.7) or Equation (2.3.8), the resulting sentence selection search algorithm over n-gram features has a complexity in $O(|\,\mathcal{C}\,| \cdot I)$.

Algorithm 2.2: Gradient ascent search.

input : $\mathbf{f}$ (source language sentence)
$\mathbf{e}_0$ (initial hypothesis)
Σ (target language vocabulary)
I (maximum translation length)
output : $\hat{\mathbf{e}}, \hat{q}$ (best translation along with its score)
auxiliary : $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$ (returns the expected BLEU score of translation $\mathbf{e}$)
$\mathrm{sub}(\mathbf{e}, \mathrm{e}, i)$ (substitutes the i^{th} word of $\mathbf{e}$ by word e)
$\mathrm{del}(\mathbf{e}, i)$ (deletes the i^{th} word of $\mathbf{e}$)
$\mathrm{ins}(\mathbf{e}, \mathrm{e}, i)$ (inserts word e as the i^{th} word of $\mathbf{e}$)

```
1  begin
2      ê ← e_0;
3      repeat
4          e_cur ← ê;
5          for 1 ≤ i ≤ |e_cur| do
6              ê_sub ← e_cur;
7              ê_ins ← e_cur;
8              for e ∈ Σ do
9                  e_sub ← sub(e_cur, e, i);
10                 if R(e_sub | f) ≥ R(ê_sub | f) then
11                     ê_sub ← e_sub;
12                 e_ins ← ins(e_cur, e, i);
13                 if R(e_ins | f) ≥ R(ê_ins | f) then
14                     ê_ins ← e_ins;
15             ê_del ← del(e_cur, i);
16             ê ← arg max_{e' ∈ {e_cur, ê_sub, ê_ins, ê_del}} R(e' | f);
17     until (R(ê | f) ≤ R(e_cur | f)) || (|ê| ≥ I);
18     return e_cur, R(e_cur | f);
```

2.4.2 Greedy Gradient Ascent Search Algorithm

Sentence selection search algorithms are limited by the fact that its search space is restricted to a finite set $\mathcal{C}$ of most probable translations. The output of these algorithms can be considered as an approximate solution that can be improved towards the "true" optimal solution. This is the rationale followed by greedy search algorithms that first automatically generates a complete solution which is iteratively improved by the application of different operators. In our case, since solutions are target language sentences, the operators used to modify the initial translation are edit operations: substitution, insertion, and deletion of single words. We want to obtain the translation of maximum score

according to Equation (2.2.3), thus we measure the improvement towards the optimal solution by the variation in the expected BLEU score.

The proposed gradient ascent search algorithm takes as input an initial translation that is iteratively edited until there is no improvement in expected BLEU score. This iterative process is based on the algorithm described in [Martínez-Hinarejos et al., 2003] to compute the median string of a set of strings[b] [Kohonen, 1985]. Algorithm 2.2 shows the pseudo-code of this greedy uphill search procedure. It takes as input a source language sentence $\mathbf{f}$, an initial translation $\mathbf{e}_0$ for $\mathbf{f}$, the target language vocabulary Σ, and the maximum length, I, of the translation. The maximum translation length sets an upper bound to the number iterations of the algorithm. As will be explained in the experiments, this upper bound is required to assure the convergence of the algorithm. While there is a possible improvement in the expected BLEU score, we modify the current solution $\mathbf{e}_{\text{cur}}$ by applying all single-word edit operations at each position i of $\mathbf{e}_{\text{cur}}$: substitution to the i^{th} word of $\mathbf{e}_{\text{cur}}$ by each word $\mathrm{e} \in \Sigma$, insertion of each word $\mathrm{e} \in \Sigma$ in the i^{th} position of $\mathbf{e}_{\text{cur}}$, and deletion of the i^{th} word of $\mathbf{e}_{\text{cur}}$. If the expected BLEU score of any of the new edited translations is higher than the score of the current translation, we repeat the process with this new improved translation $\hat{\mathbf{e}}$. In other case, we return the current hypothesis and its expected BLEU score. The output of the algorithm is a possibly new translation with a expected BLEU score higher or equal than the score of the original translation. However, since the function being optimized (Equation (2.2.3)) is not assured to be convex there is not guarantee that the final output is the globally optimal consensus translation.

Note that any target language sentence can be used as initial hypothesis, however the editions required to transform the initial hypothesis into the final consensus translation influence the temporal complexity of the algorithm. Therefore, in our experiments we used as initial hypothesis the most probable automatic translation generated by the best BLEU-scoring SMT model.

The complexity of the main for loop is $O(I \cdot |\Sigma| \cdot Z)$ (lines 5–16), where $O(Z)$ is the cost of computing $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$. The final complexity of each iteration of the algorithm would be $O(I^2 \cdot |\Sigma| \cdot |\mathcal{C}|)$ if we use the exact MBR formulation in Equation (2.2.3), and $O(I^2 \cdot |\Sigma|)$ if we use linear BLEU in Equation (2.3.7) or BLEU over features in Equation (2.3.8). Usually only a moderate number of iterations (< 10) is required for convergence.

[b] The median string $\hat{\mathbf{e}}$ of a set $\mathcal{C}$ is the string that minimizes the sum of distances, for a given distance function $\mathrm{d}(\mathbf{e}, \mathbf{e}')$, to the strings in the set: $\hat{\mathbf{e}} = \arg\min_{\mathbf{e} \in \mathcal{E}} \sum_{\mathbf{e}' \in \mathcal{C}} \mathrm{d}(\mathbf{e}, \mathbf{e}')$

P(**e** | **f**)	**e**
0.35	we are certainly faced with enormous challenges .
0.25	certainly we must tackle enormous challenges .
0.40	we are faced with enormous challenges .

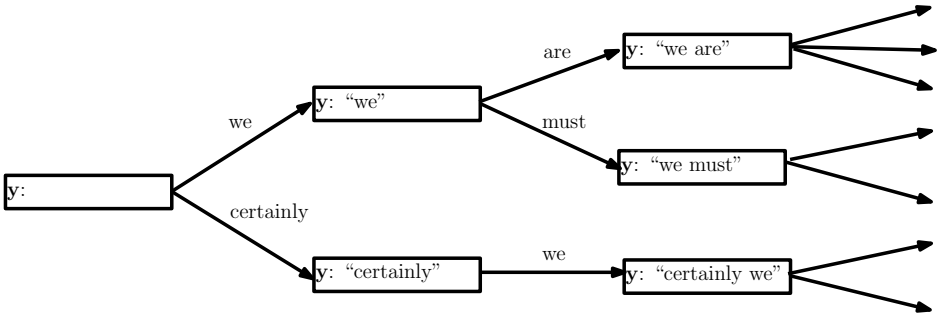


Figure 2.1: Example of the graph that represents the partial translations explored by the dynamic programming search when combining three translations.

2.4.3 Dynamic-Programming-Based Search Algorithms

Although capable to generate higher-scoring translations than the provided candidates in $\mathcal{C}$, the performance of the greedy uphill search is sensitive to the initial input translation and prone to get stuck in local optima. A more sophisticated solution to the search problem is to formalize it as a *dynamic programming* (DP) problem [Bellman, 1957].

We can interpret the search problem as a sequence of decisions that incrementally generate new translation hypotheses $\mathbf{e}'$. Starting with an empty hypothesis, each decision expand a hypotheses of size $i-1$ with one new target vocabulary word $\mathrm{e} \in \Sigma$ to create a hypothesis of size i. This search space can be represented as a directed acyclic graph where the states denote partial hypotheses and the edges are labeled with expansion words. Figure 2.1 shows an example of the two first expansions in the search graph when combining three sentences. We avoid repeated computations by traversing the search graph in a topological order, thus performing a breadth-first exploration of the search space. In other words, before we process a node, i.e. expand a hypothesis, we have to make sure that we have visited all predecessor states. We can easily guarantee the topological order by processing the nodes according to the size of the partial hypotheses.

Each possible expansion of a partial hypothesis will be assigned a score

representing its expected BLEU score. Among all possible paths of the search graph, we are interested in that of the highest score. As have been explained above, a state of the graph represents a partial hypothesis, however only the n-grams counts of the partial hypothesis are required to compute its score. Two partial hypotheses sharing the same n-grams are indistinguishable, and we are only interested in the hypothesis of higher score. According to these considerations, the search graph defined above can be simplified taking into account the particular loss function used. Since the formulation of the BLEU score (Equation (1.6.1)) includes count clippings ($\min(\cdot)$ functions in Equation (1.6.2)), we have to keep track of the exact count of each n-gram in the hypothesis. Thus, states in the search graph have to be represented by a specific multiset of n-grams. In contrast, linear BLEU in Equation (2.3.7) ignores count clippings, thus two hypotheses that share their last three words can be completed in the same way, no matter which other n-grams are present. Therefore, for linear BLEU the states in the search graph should be represented by a particular sequence of three words, similarly to the states in a De Bruijn graph [de Bruijn, 1946]. Next sections provide the different formulations of the DP-based search depending on the loss function used and describe efficient algorithms to implement them.

Dynamic-Programming-Based Search with BLEU

As we have said above, when using BLEU as loss function each state of the search graph can be represented by a specific bag (namely a specific multiset) $\mathcal{N}$ of n-grams. This is the case of the exact BLEU-based risk in Equation (2.2.3) and the BLEU-based risk over features in Equation (2.3.8). We define $Q(\mathcal{N}, \mathbf{e}) = q$, where q is the maximum score of a path leading from the initial state to the state $(\mathcal{N}, \mathbf{e})$, and $\mathbf{e}$ is the hypothesis defined by that path. The usage of both $\mathcal{N}$ and $\mathbf{e}$ may seem redundant, however, while $\mathcal{N}$ allows to distinguish between hypotheses, the actual ordered sequence of words $\mathbf{e}$ is required to generate subsequent expanded hypothesis. We also define $\widehat{Q} = \hat{q}$ as the final state of the optimal translation $\hat{\mathbf{e}}$. Finally, we obtain the following DP recursion equations:

$$Q(\emptyset, \text{""}) = 0 \tag{2.4.1}$$

$$Q(\mathcal{N}, \mathbf{e}) = \max_{\substack{\forall\,(\mathcal{N}_p, \mathbf{e}_p),\ \mathrm{e} \in \Sigma \\ \mathbf{e} = \mathbf{e}_p\,\mathrm{e},\ \mathcal{N} = \mathcal{N}_p \bigcup \Theta(\mathbf{e}_p, \mathrm{e})}} \mathrm{R}(\mathbf{e} \mid \mathbf{f}) \tag{2.4.2}$$

$$\widehat{Q} = \max_{\forall\,(\mathcal{N}_p, \mathbf{e}_p),\ \hat{\mathbf{e}} = \mathbf{e}_p\,\$} \mathrm{R}(\hat{\mathbf{e}} \mid \mathbf{f}) \tag{2.4.3}$$

where \$ is the end-of-sentence symbol that denotes a complete translation, and $\Theta(\mathbf{e}_p, \mathrm{e})$ returns the new n-grams generated when expanding hypothesis $\mathbf{e}_p$ with word e. Given a hypothesis $\mathbf{e}_p$ and an expansion word e, the expanded hypothesis $\mathbf{e} = \mathbf{e}_p\,\mathrm{e}$ contains four n-grams more than $\mathbf{e}_p$. For example, given the hypothesis $\mathbf{e}_p$="we are faced with" and the expansion word e="enormous", the expanded hypothesis $\mathbf{e}$="we are faced with enormous" contains four additional n-grams: "enormous", "with enormous", "faced with enormous", and "are faced with enormous".

As defined in the DP equations, every target language word is a potential expansion option for each partial translation. However, not all word sequences constitute correct natural language sentences. For example, given the partial translation $\mathbf{e}_p$="we are faced with", it is clear that word e="enormous" is a valid expansion option while word e="with" is not. Thus, we consider $\mathrm{e} \in \Sigma \cup \{\$\}$ as a valid expansion word for partial hypothesis $\mathbf{e}_p$ only if at least one of the new n-grams (excluding unigram e) of the resulting expanded hypothesis $\mathbf{e} = \mathbf{e}_p\,\mathrm{e}$ has a expected n-gram feature above zero. Formally, the set of expansion words $\Delta(\mathbf{e}_p)$ for a partial hypothesis $\mathbf{e}_p$ is finally computed as:

$$\Delta(\mathbf{e}_p) = \left\{ \mathrm{e} \in \Sigma \cup \{\$\} \;\mid\; \exists\, \mathbf{w} \in \Theta(\mathbf{e}_p, \mathrm{e}) \setminus \{\mathrm{e}\} \wedge \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\phi_{\mathbf{w}}(\mathbf{e}')] > 0 \right\}$$

To solve the DP search described above, we have to compute the expected BLEU score $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$ for each possible state and expansion word[c]. Unfortunately, the number of states, that can be computed by the multiset coefficient [Stanley, 2002], is factorial in the size of the target vocabulary Σ. Therefore, we cannot expect to efficiently obtain the optimal solution. To speed up the search, we use a *beam search* strategy [Jelinek, 1997]. The idea of beam search is that at each step, we expand only the most promising hypotheses and discard hypotheses that are unlikely to lead to the optimal solution. In contrast to the conventional DP search, beam search may result in suboptimal solutions. Specifically, we apply pruning to reduce the number of expansion words and states.

Regarding the states in the search graph, we can impose a limitation in the maximum number of them that are expanded. At each step of the graph exploration, we expand only the N best-scoring states and discard the rest of them. Note that during pruning, we compare hypotheses which contain different n-grams. Here, it is important to use a rest score estimate for completing the hypothesis. Without such a rest score estimate, the search would focus on hypotheses that contain high-probability n-grams at the beginning even if

[c]The expected BLEU score cannot be computed incrementally due to the $\min(\cdot)$ functions in its formulation.

Algorithm 2.3: Dynamic programming beam search for BLEU.

input : $\mathbf{f}$ (source language sentence)
N (pruning parameter)
I (maximum translation length)
output : $\hat{\mathbf{e}}, \hat{q}$ (optimal translation along with its score)
auxiliary : $\Theta(\mathbf{e}, \mathrm{e})$ (new n-grams after expanding hypothesis $\mathbf{e}$ with word e)
$\Delta(\mathbf{e})$ (set of expansion words for hypothesis $\mathbf{e}$)
$\overline{\mathrm{R}}(\mathbf{e} \mid \mathbf{f})$ (returns the complete score of $\mathbf{e}$)
$\Pi(i, N)$ (returns the states of size i that remain after pruning)

```
1  begin
2      Q(∅,"") ← 0;  ê ← "";  Q̂ ← 0;
3      for i = 1 to I do
4          forall the (N_p, e_p) : |e_p| == i − 1 ∧ (N_p, e_p) ∈ Π(i − 1, N) do
5              forall the e ∈ Δ(e_p) do
6                  e ← e_p e;  q ← R̄(e | f);
7                  if e == $ then
8                      q̂ ← Q̂;
9                      if q > q̂ then
10                         ê ← e;  Q̂ ← q;
11                 else
12                     N ← N_p ⋃ Θ(e_p, e);
13                     q' ← Q(N, ·);
14                     if q > q' then
15                         Q(N, e) ← q;
16     return ê, q̂;
```

they cannot be extended to complete a full translation. This is of course undesirable. To assure a fair competition between hypotheses, we follow the ideas in [He and Toutanova, 2009]. We apply a light search process (considering at each step only the single best state expansion) to estimate the expected BLEU score of the complete translation that may be obtained from the hypothesis. This score $\overline{\mathrm{R}}(\mathbf{e} \mid \mathbf{f})$ is then used as the complete score of the hypothesis.

Algorithm 2.3 shows the pseudo-code of the DP beam search with pruning. It takes as input the source language sentence ($\mathbf{f}$), the number of hypotheses to keep after pruning (N), and the maximum translation length under consideration (I). We use some auxiliary functions: $\Theta(\mathbf{e}, \mathrm{e})$ returns the set of new n-grams generated in the expansion of hypothesis $\mathbf{e}$ with word e, $\Delta(\mathbf{e})$ returns the set of valid expansion words for $\mathbf{e}$, $\overline{\mathrm{R}}(\mathbf{e} \mid \mathbf{f})$ returns the complete score

(current score plus rest score estimation) of $\mathbf{e}$, and $\Pi(i, N)$ is a function that returns the N best-scoring states representing partial hypotheses of size i; lower-scoring states are pruned out.

The first loop in Algorithm 2.3 assures that the search graph is traversed in topological order. Additionally, it introduces an upper bound to the maximum translation size under consideration, and thus, to the number of iterations of the algorithm. At each iteration, line 4 loops over the non-pruned states that store a translation of size $i-1$. For each of these predecessor states, line 5 loops over the corresponding expansion words. Given a predecessor state $(\mathcal{N}_p, \mathbf{e}_p)$, and a valid expansion word e, we compute the complete score (current score plus rest score estimation) q of the expanded hypothesis $\mathbf{e} = \mathbf{e}_p\,\mathrm{e}$ (line 6). Then, if the expansion word is the end-of-sentence symbol (e == \$), the expanded hypothesis is a complete translation, and if it improves the score $\hat{q}$ of the best consensus translation so far, we update the current optimal consensus translation and its score (lines 7–10). For any other expansion word, we first compute the bag of n-grams $\mathcal{N}$ of the expanded hypothesis (line 12). Then, if the score q of the expanded hypothesis improves the score, q', stored in the corresponding successor state $(\mathcal{N}, \cdot)$ (line 14), we update the state.

This beam search algorithm with pruning has a computational complexity in $O(I{\cdot}N{\cdot}D{\cdot}Z)$, where I is the maximum translation length in line 3, N denotes the pruning parameter that controls the maximum number of predecessor states in line 4, D denotes the maximum number of expansion words in line 5, and $O(Z)$ is the computational complexity of computing the expected BLEU score in line 6. The final cost of the algorithm would be: $O(I^2 \cdot N \cdot D \cdot |\,\mathcal{E}\,|)$ if the exact sentence-wise risk computation (Equation (2.2.3)) is chosen, or $O(I^2 \cdot N \cdot D)$ if we choose BLEU-based risk over features in Equation (2.3.8).

Dynamic Programming Search with Linear BLEU

Using linear BLEU, two partial hypotheses that share their last three words also share their optimal expansion up to a complete translation. Therefore, the states in the search graph can be represented as a particular sequence of three words σ. All partial hypothesis arriving to a particular state will share the same history σ, To distinguish between hypotheses of different size, search states are also indexed by the size of the hypotheses that arrive to the state. Similarly as done for BLEU, we define the quantity $Q(i, \sigma)$ to denote the maximum score of a path leading from the initial state to the state (i, σ). We also define $\widehat{Q}$ as the score of the optimal translation $\hat{\mathbf{e}}$. Finally, we obtain the

Algorithm 2.4: Dynamic programming search for linear BLEU.

input : $\mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')]$ (indicator feature expectations)
$\lambda_0, \lambda_{\mathbf{w}}$ (values of linear BLEU free parameters)
I (maximum translation length)
output : $Q(i, \sigma)$ (search graph)
$B(i, \sigma)$ (back-pointer to the best predecessor state)
auxiliary : tail($\mathbf{e}$) (returns the last three words of word sequence $\mathbf{e}$)
$\Theta(\mathbf{e}, \mathrm{e})$ (new n-grams after expanding hypothesis $\mathbf{e}$ with word e)
$\Delta(\mathbf{e})$ (set of expansion words for hypothesis $\mathbf{e}$)

```
1  begin
2      Q(·,·) ← 0;
3      for i = 1 to I do
4          forall the σ_p ∈ Q(i−1,·) do
5              forall the e ∈ Δ(σ_p) do
6                  q ← Q(i−1,σ_p) + λ_0 + Σ_{w∈Θ(σ_p,e)} λ_w · E_{P(e'|f)}[δ_w(e')];
7                  σ ← tail(σ_p e);
8                  if q > Q(i,σ) then
9                      Q(i,σ) ← q;
10                     B(i,σ) ← (i−1,σ_p);
```

following DP recursion equations:

$$Q(0,"") = 0 \tag{2.4.4}$$

$$Q(i,\sigma) = \max_{\substack{\mathrm{e}\in\Sigma \\ q_p = Q(i-1,\sigma_p) \\ \sigma = \mathrm{tail}(\sigma_p\,\mathrm{e})}} \left\{ q_p + \lambda_0 + \sum_{\mathbf{w}\in\Theta(\sigma_p,\mathrm{e})} \lambda_{\mathbf{w}} \cdot \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')] \right\} \tag{2.4.5}$$

$$\widehat{Q} = \max_{\substack{q_p = Q(\cdot,\sigma_p) \\ \sigma = \mathrm{tail}(\sigma_p\,\$)}} \left\{ q_p + \lambda_0 + \sum_{\mathbf{w}\in\Theta(\sigma_p,\$)} \lambda_{\mathbf{w}} \cdot \mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')] \right\} \tag{2.4.6}$$

where $\lambda_0, \lambda_{\mathbf{w}}$ are the free parameters in Equation (2.3.7), tail(σ e) returns the last three words of word sequence σ e, and $\Theta(\sigma, \mathrm{e})$ returns the new n-grams generated when adding word e at to sequence σ.

Since the number of states is at most cubical with the output vocabulary, we can implement it exactly without the need for beam search. Algorithm 2.4 shows the pseudo-code of the DP search for linear BLEU. It takes as input the indicator feature expectations $\mathbb{E}_{\mathrm{P}(\mathbf{e}'|\mathbf{f})}[\delta_{\mathbf{w}}(\mathbf{e}')]$, the values of the free parameters $\lambda_0, \lambda_{\mathbf{w}}$ in the linear BLEU formulation, and the maximum translation

length under consideration I. At each iteration the algorithm loops over the predecessor states (line 4), and the corresponding expansion words (line 5) for each state. Given a predecessor state $(i-1, \sigma_p)$, we compute the score, q, of the expanded hypothesis (line 6), and if the score q of the expanded hypothesis improves the score stored in the corresponding successor state (i, σ) (line 8), we update the state and the corresponding back-pointer $B(i, \sigma)$. Once the search process has been completed, back-pointer variables allow us to retrieve the target sentence of highest probability.

This DP search algorithm has a computational complexity in $O(I \cdot |\Sigma|^3 \cdot D)$, where I is the maximum translation length in line 3, $|\Sigma|$ is the size of the output vocabulary that controls the maximum number of predecessor states in line 4, and D denotes the maximum number of expansion words in line 5.

2.5 Experiments

We now describe the experiments performed to study the soundness of the system combination method, MBRSC, proposed above in this chapter.

First, we conducted several comparative experiments where we studied the performance of the different risk computation methods and search algorithms. The objective of this experimentation is to determine the combination of risk computation method and search algorithm, namely the MBRSC setup, that generates consensus translations of higher quality. Results for these experiments are shown in Section 2.5.1. Then, we carried out a second set of experiments to compare the performance of the best MBRSC setup with several state-of-the-art system combination methods. We present results for this experimentation in Section 2.5.2.

2.5.1 Comparative Experiments

As we have described in Section 2.2, the exact formalism of the MBRSC method in Equation (2.2.3) cannot be exactly implemented due to its high computational complexity. To efficiently implement MBRSC, we split Equation (2.2.3) into two subproblems (risk computation and search) and give several solutions to deal with each of them in Sections 2.3 and 2.4:

Risk computation methods	Search algorithms
exact BLEU, Eq. (2.3.1)	sentence selection, Alg. 2.1
linear BLEU, Eq. (2.3.7)	gradient ascent, Alg. 2.2
BLEU over expectations, Eq. (2.3.8)	DP beam, Alg. 2.3 and 2.4

Corpus	Sentences	Tokens (Fre/Eng)	Vocabulary (Fre/Eng)
Europarl	1.4M	44.7M/40.0M	129.2k/107.7k
Gigaword	22.5M	811.2M/668.4M	2.7M/2.9M
News Commentary	64.2k	1.8M/1.6M	46.1k/38.8k

Table 2.1: Main figures of the French-English training corpora provided in the 2009 Workshop on Statistical Machine Translation. These corpora were use by the participants in the workshop to build their translation models. In our experiments, we combine the automatic translations submitted to the workshop by the different research groups. M and k denote millions and thousands of elements respectively.

In this section, we show the results of the experiments carried out to evaluate the performance of these solutions. On each experiment we generated consensus translations using a particular combination of a risk computation method and a search algorithm. There are nine possible combinations, but due to the high computational complexity of the exact BLEU risk computation, we used it only jointly with sentence selection search. We chose this particular configuration, sentence selection search with exact BLEU risk, as the baseline in our experiments. This left seven different combinations, or setups, that were tested in the experiments. The final goal of the experimentation was to determine the combination that generates consensus translations of higher quality, namely the optimal MBRSC setup.

The experiments were performed on French-English, from the translation task of the 2009 Workshop on Statistical Machine Translation[d] [Callison-Burch et al., 2009]. Table 2.1 displays the main figures of the different training corpora used by the participants to build their translation models. Then, the participants were asked to provide automatic translations into English for news articles that were drawn from a variety of sources. A total of 136 articles were selected to be translated consisting of roughly $80,000$ french words across 3027 sentences. The translated articles were split into a development corpus and a test corpus containing 502 sentences ($\sim 13,000$ words) and 2525 sentences ($\sim 66,000$ words) respectively.

In the experiments, we combined the outputs of the five statistical MT systems that submitted lists of N-best translation options to the task. Table 2.2 shows the average number of alternative English translations for each French sentence, and case insensitive BLEU scores for the single best translation of

[d] `http://statmt.org/wmt09/translation-task.html`

System	Development		Test	
	#trans_opts	BLEU [%]	#trans_opts	BLEU [%]
A	13	26.5	13	24.8
B	9	26.8	9	25.2
C	256	27.1	263	25.8
D	127	27.2	126	26.4
E	40	27.5	41	25.8

Table 2.2: Average number of translation options provided, and case insensitive BLEU scores for the single best translation of each system. These are the translations submitted by five of the participants in the French-English shared translation task of the 2009 Workshop on Statistical Machine Translation. Participants use the corpora described in Table 2.1 to train their MT models.

each system. System outputs were tokenized and lower-cased before performing the combination. We report case-insensitive evaluation results to factor out the effect of true-casing of the English words from the effect of computing a consensus translation.

Preliminary Experiments

DP beam search using BLEU risk over n-gram count expectations (Algorithm 2.3) implements pruning to deal with the exponential number of states that have to be explored through decoding. Thus, we carried out a preliminary series of experiments to study how the value of the pruning parameter N, namely the number of hypotheses kept after pruning, affects the performance of Algorithm 2.3 in terms of translation quality and decoding time[e]. Figure 2.2 displays the quality of the generated consensus translations (on the left vertical axis) and the total decoding time (on the right vertical axis) as functions of N. We observed that decoding time increased linearly with N (note that N is log-scaled in Figure 2.2) while the quality of the consensus translations stayed approximately constant with only slight improvements for larger N values. Despite the scarce quality difference, the score of the generated consensus translations did improve as larger N values were considered which indicates that larger values of N allowed us to obtain better scoring consensus translations.

Given these results, we considered that a pruning parameter value of $N = 10$ provided the optimal trade-off between translation quality and de-

[e]Our test machine is a PC with an Intel Core© i5-3570K processor running at 3.40 GHz.

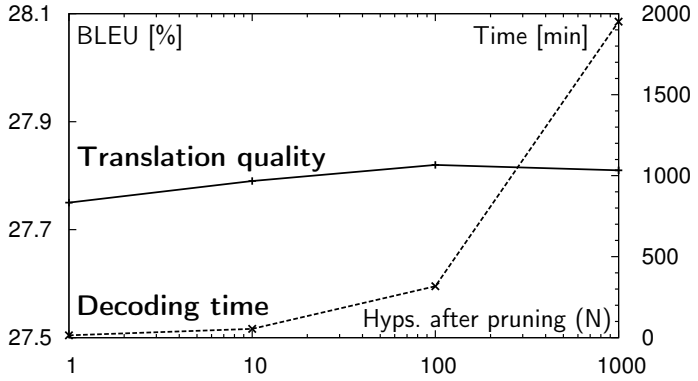


Figure 2.2: BLEU score of the consensus translations (on the left vertical axis) and total decoding time (on the right vertical axis) obtained by the DP beam decoding algorithm using BLEU risk over expected n-gram counts (Algorithm 2.3) as a function of the number of hypotheses kept after pruning (N).

coding time. Thus, this is the value used in the experimentation. A further study of the influence of this value is presented in the following sections.

Results on MBRSC Setups

Table 2.3 displays case-insensitive BLEU and TER results for the computed consensus translations generated by different setups of MBRSC. Results were computed for the test corpus. We used the development corpus to compute the values of the parameters $(\lambda_0, \lambda_{\mathbf{w}})$ in the linear BLEU formulation, Equation (2.3.5). All experiments were carried out using uniform ensemble weights $\boldsymbol{\alpha}$ in Equation (2.2.1). This approach allowed us to fairly compare the different MBRSC setups factoring out possible influences of the parameter optimization process. The maximum translation length I under consideration was always equal to the length of the longest translation option. For each source sentence, we combined all the translation options provided by the five individual systems. On average, about 450 English translations were combined for each french sentence. Additionally, we report results for the best and worst individual systems.

As a first (baseline) experiment, we present results for the sentence selection search algorithm with exact BLEU risk computation. The risk of each candidate translation was computed by exhaustively calculating its BLEU score with respect to the rest of the translations as in Equation (2.3.1) and the best-scoring candidate translation was selected as the final consensus translation.

System setup		BLEU[%]	TER[%]
worst single system		24.8	60.4
best single system		26.4	56.0
sentence selection	exact BLEU (baseline)	27.4***	55.5
	linear BLEU	27.2	56.2
	BLEU over expectations	27.4***	55.5
gradient ascent	linear BLEU	26.3	59.6
	BLEU over expectations	27.7***	55.4
DP beam	linear BLEU	26.8	57.8
	BLEU over expectations	27.8***	55.1

Table 2.3: Quality of the consensus translations generated by different MBRSC setups. All experiments used uniform values for the ensemble weights. For each search algorithm, asterisks denote a statistically significant difference in BLEU with respect to linear BLEU (99% confidence).

Results in Table 2.3 show that this baseline already resulted in a substantial improvement over the best individual system: +1.0 BLEU points and −0.5 TER points.

We replicated this baseline sentence selection experiment using the two proposed alternatives to the exact BLEU risk computation: linear BLEU, and BLEU over n-gram count expectations. With this experiment we aimed at estimating the accuracy of the proposed alternatives. On the one hand, the use of linear BLEU risk resulted in a worse −0.2 points BLEU score, and a degradation of +0.7 points in TER. Although scarce, this differences were statistically significant. On the other hand, the use of BLEU over n-gram count expectations obtained the same BLEU and TER scores than the exact BLEU risk with no statistically significant difference. These results indicate that BLEU risk over n-gram count expectations is a pretty accurate approximation to the exact MBR classifier even for nonlinear loss functions such as BLEU, a finding consistent with prior research [DeNero et al., 2009]. The performance degradation obtained when using linear BLEU risk can be attributed to the lack of n-gram count clippings in its formulation in Equation (2.3.7). This resulted in longer consensus translations (26.8 words on average) than the ones selected when using exact BLEU risk (26.5 words) or BLEU risk over n-gram count expectations (26.5 words), and also longer than the average length (26.0 words) of the reference translations.

Next, we generated consensus translations using the greedy gradient ascent

search algorithm described in Section 2.4.2. Results for linear BLEU showed an important degradation in performance (−0.9 BLEU points and +3.4 TER points) with respect to sentence selection search. In contrast, results for BLEU over expectations slightly improve sentence selection search results by +0.3 BLEU points and −0.1 TER points. Again performance for BLEU risk over expectations were statistically better than results for linear BLEU risk. Since the outputs of the gradient ascent algorithm are assured to have a score higher or equal than the input translation, we conclude that BLEU over features allows to correctly score the new candidate translations explored by the search algorithm while linear BLEU have problems in dealing with them.

The explanation for the low performance of linear BLEU stems again in the lack of count clippings in its formulation. As the algorithm perturbs the current solution, new words that form highly probable n-grams may be repeatedly added to the current translation. This phenomena seems to be very common in this corpus: generated consensus translations are much longer (27.8 words on average) than the reference (26.0 words), and about 24% of the generated consensus translations reach the maximum translation length[f]. A similar phenomena occurs when the initial translation contains infrequent n-grams, the algorithm may keep deleting words until only a few highly probable n-grams remain. Table 2.4 shows various examples of these consensus translations. The two first are examples of consensus translations with repeated highly probable n-grams while the last two examples are sentences where infrequent n-grams have been deleted. Here, the consensus translations are compared with the input translations passed to the gradient ascent search, Algorithm 2.2.

Then, we generated consensus translations using the DP beam search algorithms described in Section 2.4.3. As in previous experiments, BLEU over expectations obtains a statistically significant improvement in performance over linear BLEU. Results for linear BLEU improved the results for gradient ascent search (+0.5 BLEU points and −1.8 TER points), but they are still quite below the results for the simpler sentence selection search. Regarding BLEU risk over features, results showed a further slight performance improvement (+0.1 BLEU points and −0.4 TER points) over gradient ascent search for a total of +0.4 BLEU points and −0.4 TER points over sentence selection search.

Finally, we also compared the three search algorithms under study, sentence selection (Algorithm 2.1), gradient ascent (Algorithm 2.2) and DP beam (Algorithms 2.3 and 2.4), in terms of decoding time. We estimate decoding

[f]Without the maximum length restriction the gradient ascent search will not converge for these translations.

I:	(1.5)	" we have made great progress . "
C:	(3.3)	" we have made great progress . " *we have made*
R:	(——)	" we 've made great progress .
I:	(14.0)	i am curious to know if i could see here .
C:	(22.7)	*am curious to know if* i am curious to know if i could see here .
R:	(——)	i 'm looking forward to finding out whether that 's happening here , too .
I:	(−1.7)	fall actions in asia
C:	(−0.6)	asia
R:	(——)	stocks fall in asia
I:	(−1.9)	no current apparatus is as the telephone .
C:	(−0.4)	that the telephone .
R:	(——)	no contemporary machine is as universal as the telephone .

Table 2.4: Examples of consensus translations (**C**) generated by the gradient ascent algorithm with linear BLEU risk. We also display the initial translations (**I**) passed to the gradient ascent algorithm and the corresponding reference translations (**R**). These erroneous consensus translations are result of the lack of n-gram count clippings in the linear BLEU formulation. Linear BLEU expected scores are given in parenthesis.

time by the number of times each algorithm calls the risk-computation function $\mathrm{R}(\mathbf{e} \mid \mathbf{f})$ during the generation of consensus translations for the whole test corpus. We report this count instead of the actual decoding time because it is independent of the selected risk function, which, we think, allows us to fairly compare the complexity of the different algorithms. We observed that sentence selection made ~ 1.1 million calls to the risk function, while gradient ascent made ~ 23 million calls to the risk function, and DP decoding made ~ 15 million calls including those involved in the estimation of the rest score. It is worthy of notice that in our experiments the number of translations to be combined on average was moderate (450) which explains the low computational complexity of sentence selection. However, in scenarios with a larger number of translations, sentence selection search can rapidly become intractable.

Regarding the different risk functions, both linear BLEU and BLEU over expectations have constant complexity which allows us to implement them in all three search algorithms. In contrast, the complexity of the exact BLEU risk depends on the number of translations being combined (see Equation 2.3.1). We observed this fact in the experiments where the exact BLEU risk could only be effectively applied to sentence selection search. As an example, total

System setup	BLEU[%]	TER[%]
best single system	26.4	56.0
sentence selection	27.6	55.2
gradient ascent	27.8***	55.3
DP beam	28.0**	54.9
oracle (DP beam + reference n-gram counts)	43.3***	42.2

Table 2.5: Quality of the consensus translations generated by different search algorithms using BLEU risk over expectations. Ensemble weights were tuned to optimize BLEU in the separate development set. Oracle denotes the upper bound of the performance for the DP beam search algorithm. Asterisks denote the minimum statistical significance of each system with respect to the systems above, ** 95%, *** 99% confidence.

decoding time for DP beam search using BLEU over expectations was ~ 55 minutes (~ 1.3 seconds per sentence) while the estimated time using sentence selection was ~ 17 days (~ 9.5 minutes per sentence).

Up to this point, we can already extract some conclusions on the different risk computation methods:

- Linear BLEU is a good approximation to the exact BLEU for sentence selection search, but due to the lack of n-gram count clippings in its formulation, it fails at scoring the new translations explored by gradient ascent or DP beam search algorithms.

- BLEU over expectations performs as the exact BLEU risk for sentence selection search, and it effectively scores new translations explored by gradient ascent search and DP beam search. As a result the use of BLEU over features allows to generate improved consensus translations that may be different than the provided translation options.

- Computational complexity limits the application of exact BLEU risk to the simpler sentence selection search algorithms.

According to these considerations BLEU over expectations provides a well balanced tradeoff between efficiency and performance, showing a consistent statistically significant better performance than linear BLEU for all search algorithms. Thus, although BLEU risk over expectations has a higher computational complexity than linear BLEU, it is the method chosen to compute the risk in the rest of the experimentation. Regarding the search algorithms, DP

beam seemed to be the best performing algorithm. However, since BLEU and TER differences were scarce, we performed a further study on the different search algorithms to determine which one provides better performance.

Further Results on Search Algorithms

Table 2.5 displays the performance of the different search algorithms using BLEU risk over expectations. We trained the values of free parameters $\boldsymbol{\alpha}$ of the ensemble so that they optimize BLEU in a separate development set. Results show that each subsequence combination method outperformed the methods listed above it in the table. Gradient ascent search improved sentence selection by +0.2 BLEU, although a slight degradation in TER was also observed. DP beam search further improved performance of gradient ascent: +0.3 BLEU points and −0.4 TER points. Again, despite slight, these differences in performance were statistically significant.

We performed one last comparative experiment (oracle) to measure the upper bound for the performance of DP beam search. For the oracle system, instead of n-gram count expectations, we generated consensus translations using n-gram counts computed directly from the reference translations. Naturally, oracle results showed a huge improvement in performance over the best individual system. Since DP beam search explores about one tenth of this potential, we conclude that refinements in the estimation of n-gram count expectations could have the potential to boost translation quality.

Results in Table 2.5 showed that using uniform weights there was no statistically significant difference between gradient ascent and DP beam search, only by optimizing the parameters we were able to obtain a statistically significant improvement for DP beam search. Thus, to perform a fair comparison between the search algorithms, we automatically optimized the values of free parameters $\boldsymbol{\alpha}$ in a separate development set for each algorithm. Then, we studied the differences in performance as we varied the number of translation options combined. For each source sentence only a subset of translation options are combined to generate the consensus translation, namely the top scoring ones. Figure 2.3 compares the BLEU score of the consensus translations generated by sentence selection (SS), gradient ascent (GA), and DP beam (DP) as a function of the number of translation options combined. Additionally, we report significance levels of the pairwise differences in performance[g]. We mark two standard levels of significance, 0.01 and 0.05, for reference.

[g]Similarly as done in [Becker, 2008], we give p-values on a logarithmic scale. Note that 10^{-4} is the smallest possible p-value that can be computed with $10,000$ shuffles.

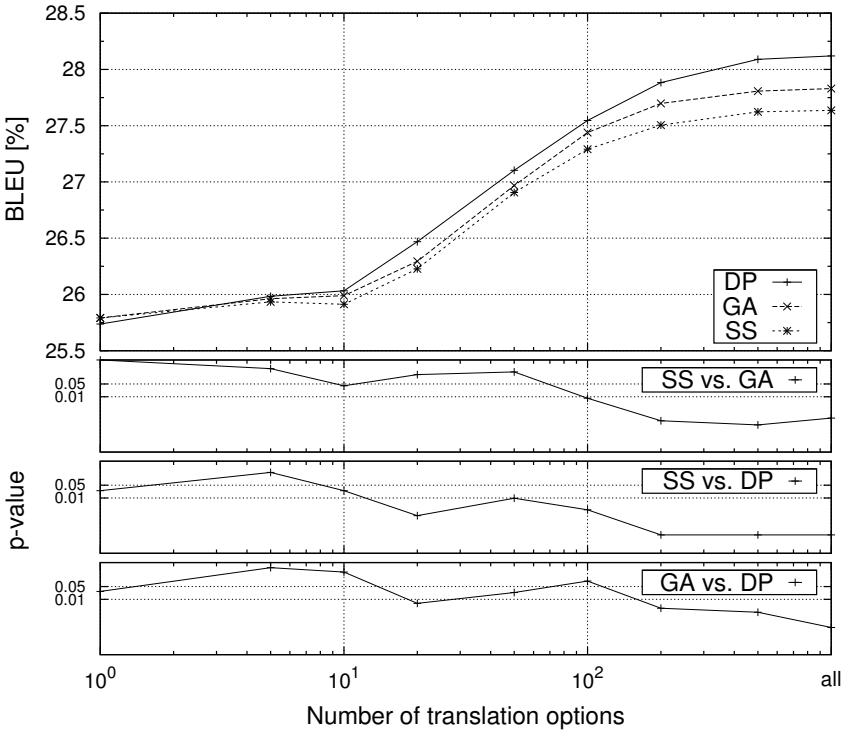


Figure 2.3: BLEU scores, and statistical significance of the pairwise differences between sentence selection search (SS), gradient ascent search (GA), and DP beam search (DP). Parameter optimization was performed individually for each search algorithm.

BLEU results in the first panel of Figure 2.3 show that DP beam search consistently outperformed gradient ascent search, and that gradient ascent search consistently outperformed sentence selection search. According to the second panel in the figure, these BLEU differences between sentence selection and gradient ascent were significant only when 100 or more translation options were combined. However, the performance differences between sentence selection and DP beam search were statistically significant from only 10 translation options (third panel). Finally, more than 100 translation options had to be combined to found a statistically significant difference in BLEU between gradient ascent and DP beam search (fourth panel). These results were no surprising since the search space for sentence-selection search grows linearly with the number of translation options while for DP beam search and gradient ascent search it grows exponentially. Thus, as more translation options were

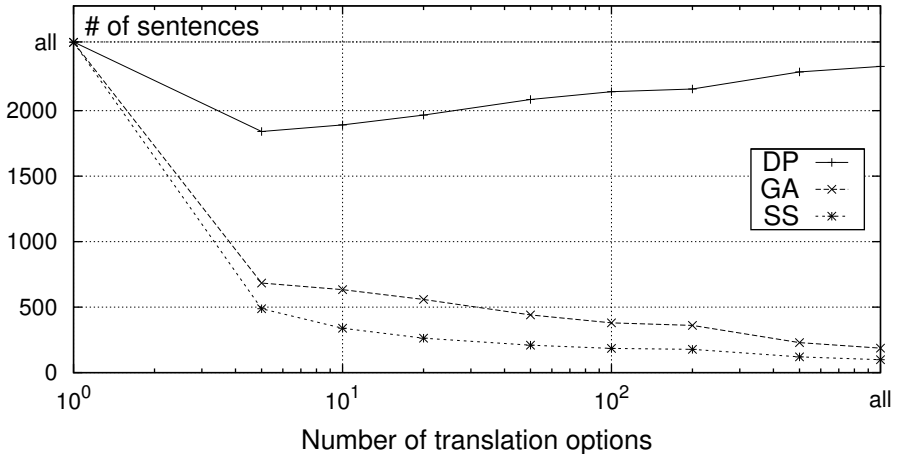


Figure 2.4: Number of sentences for which the consensus translation returned by a particular search algorithm scores higher than the consensus translations generated by the other algorithms. Ties were allowed. A total of 2525 consensus translations were generated.

used the latter two algorithms were able to explore a broader space and to find consensus translations of better quality. Regarding gradient ascent and DP beam search, when only a few translation options were combined and thus the search space was "small" both algorithms were able to successfully explore it, but as the search space grew the quality of the consensus translation generated by DP beam also improved up to the point that there was a statistically significant difference with respect to gradient ascent search.

Figure 2.4 confirms these considerations. It shows the number of times the consensus translation generated by a particular search algorithm had a higher score than the consensus translations generated by the other algorithms. Ties were allowed. We present these numbers as a function of the number of translations options combined. For most source sentences, DP beam search returned a consensus translation with higher score. Moreover, the number of sentences for which DP beam obtained the higher-scoring consensus translation grew with number of translation options. When combining one single translation all algorithms obtain the same solution and thus all of the consensus translations had the highest score.

The following conclusions can be extracted from these results:

- Sentence selection search is limited by the fact that cannot generate new translations different from the provided translation options.

single MT	no aircraft universal also today is that the telephone .
MBRSC	no current apparatus is as universal as the telephone .
reference	no contemporary machine is as universal as the telephone .
single MT	no confirmation was able to be obtained from aig .
MBRSC	no confirmation could be obtained from aig .
reference	no confirmation could be obtained from aig .
single MT	this period has been reduced to less than a week .
MBRSC	it was able to reduce this period of less than a week .
reference	it was able to reduce that period to less than a week .

Table 2.6: Examples of translation quality improvements resulting from system combination.

- Gradient ascent search explores a larger search space than sentence selection. However, only a limited number of times it is able to found a consensus translation of higher score. Nevertheless, these consensus translations boosts its performance above sentence selection.
- DP beam search effectively explores a larger search space than sentence selection. Additionally, most of the times it obtains a better higher scoring consensus translation than the other search algorithms which results in a statistically significant better performance.

Wrapping up these conclusions and the conclusions on the previous section, we can affirm that the best MBRSC setup is given by a combination of BLEU risk over expectations and DP beam search with parameter optimization. From now on, this is to be considered the standard configuration for MBRSC, and is the MBRSC setup used in the rest of the experiments. To conclude these comparative experiments. Table 2.6 shows examples of how the translation quality can be improved with system combination. Here, the consensus translation generated by MBRSC is compared with the translation of the best individual system, as well as with a human reference translation.

2.5.2 Comparison to State-of-the-art Methods

We now compare MBRSC against several state-of-the-art subsequence system combination techniques. These experiments were performed on the official evaluation sets from the system combination task [h] of the 2011 Workshop on

[h] http://www.statmt.org/wmt11/system-combination-task.html

System	cz→en	en→cz	de→en	en→de
MBRSC	29.5	20.8	25.2	18.4
BBN [Rosti et al., 2011]	29.9	–	26.5	–
CMU [Heafield and Lavie, 2011]	28.7	20.1	25.1	17.6
JHU [Xu et al., 2011]	29.4	–	24.9	–
RTWH [Leusch et al., 2011]	–	–	25.4	–

Table 2.7: BLEU [%] scores of MBRSC in comparison with the best-performing system combination methods presented in the system combination task of the 2011 workshop on statistical machine translation.

Statistical Machine Translation [Callison-Burch et al., 2011]. Consensus translations were generated for both translation directions of the following language pairs: Czech–English (cz–en), German–English (de–en), Spanish–English (es–en) and French–English (fr–en). For each translation direction, we combined the outputs of all the system that submit translations to the translation task. In contrast to the previous experiments, for each source sentence only single best translations were provided by each individual system. Thus, each experiment combined only about 10 translations.

Table 2.7 compares the performance of MBRSC with respect to the various systems that participate in the system combination task. For the sake of simplicity, we show results only for the four (out of ten) best-performing systems. All these system combination methods align the provided translations to build a consensus network, and compute the consensus translation as the highest-scoring path through the network in the style of [Fiscus, 1997]. They differ in the alignment method and the path-scoring models used. We report results only for cz↔en and de↔en translation directions. Experiments for other directions led to similar conclusions.

It is important to note that the experimental conditions of this task favored consensus network methods. On the one hand, only single-best translations were available so the n-gram count expectations could not be smoothly estimated and were biased to those single translations. On the other hand, task organizers allowed the use of any additional data which permits network methods to train their complex search models. For example, Rosti et al. [2011] used additional data that amounts for a total of $6.4 \cdot 10^9$ words. In contrast, MBRSC works directly on the provided translations, thus, its performance was not limited by the availability of additional data. Still, we found that even in this pessimistic setting MBRSC was the best performer for en→cz and en→de (although the differences were scarce), and was between the top-performing

systems for the rest of translation directions.

Not surprisingly, MBRSC scored particularly high for those translation directions (cz and de) whose target language had scarcer resources. For these languages, network-based systems simply did not had enough data to train their complex network search models. In fact, many participants submitted consensus translations for only a limited number of target languages. In contrast, MBRSC does not require any additional data. Since the consensus translation is directly computed from the provided translation options, MBRSC obtained competitive results in all translation directions. These results confirm the soundness and generality of the proposed system combination technique.

2.6 Summary

We have presented a new system combination method for MT that gathers together the advantages of sentence selection, and subsequence combination methods. We have started by introducing the system combination problem, and the particular problems that must be faced when combining MT systems.

We have formalized the MT system combination problem as a statistical classification problem. First, the probability distributions of the systems being combined have been included in a weighted ensemble to define a new combined model. We then have defined the optimal decision function for this model using the BLEU [Papineni et al., 2002] score as loss function. This optimal decision function has a high temporal complexity quadratical with the number of potential sentences in the target language. To deal with this high complexity, we have presented various approaches to efficiently perform the risk computation, and the search of the optimal translation.

We have shown that the risk computation procedure can be simplified whenever the loss function under consideration is a linear function of some reference features. Unfortunately, BLEU is not, thus we have described two approaches to achieve the desired complexity reduction. On the one hand, we have proposed to use a linear alternative to BLEU [Tromble et al., 2008]. This allows us to efficiently obtain the exact risk but using a linear alternative to BLEU, not the exact BLEU score. On the other hand, we used with the exact BLEU score as loss function, but we had to compute the risk with an alternative to the optimal decision function. Finally, we have explained how to efficiently compute the feature expectations required by both methods.

We have described the problems involved in the search for the optimal translation, and three algorithms have been presented to deal with this search problem. First, we have presented a straightforward sentence-selection search

algorithm that explicitly enumerate all candidate translations to make its decision. Then, we have described two subsequence combination methods that explore a wider search space beyond the translations provided by the individual systems. The first one is a gradient ascent search algorithm that iteratively modifies an approximate solution towards the optimal translation. The second is a DP search algorithm that allows for an efficient exploration of the full target language. Also it is less prone to get struck in local optima than the gradient ascent algorithm. Additionally, we have provided a computational complexity analysis for each of the algorithms depending on the method selected to compute the risk.

We have tested the different components of the proposed system combination method. First, we have conducted a series of comparative experiments to determine the best combination of risk computation method and search algorithm for MBRSC. Then, we have used this MBRSC setup to compare our system with several state-of-the-art system combination methods.

The comparative experiments have started by studying the performance of the different combinations. Due to the computational complexity of exact BLEU risk, it was only tested with the simpler sentence selection algorithm. Results showed that BLEU risk over expectations is clearly the best risk computation method since it obtained virtually the same results as the exact BLEU risk for sentence selection search, and statistically outperforms linear BLEU whatever the search algorithm used.

After determining the best risk computation method, we conducted further experiments to determine the best search algorithm. Results of these experiments showed that DP beam search explores a larger search space than sentence selection, and additionally did that more effectively than gradient ascent search. These facts made DP beam search able to generate statistically better consensus translations whenever an enough number of translations options were combined.

Once the best combination of risk computation method and search algorithm have been established, we have compared this standard MBRSC setup with other system combination methods. Results showed that the performance of MBRSC is on the same level than the state-of-the-art system combination methods. Moreover, since MBRSC generates the consensus translation directly from the provided translation options, its performance is not limited by the availability of additional data which makes MBRSC a particularly well suited method to be applied to languages with scarce resources.

Chapter

3 Machine Translation Quality Estimation

Although significant progress has been observed in the overall quality of MT systems in recent years, it is well known that translation quality can vary considerably across translated segments. The need for models to assess the quality of translated segments is becoming more and more evident for the efficient deployment of MT technology. We address quality estimation as a two-step regression problem where multiple features are combined to predict a quality score. First, a dimensionality reduction module extracts, from the original features, the set of variables that better explain translation quality. Then, a prediction model is built from those variables to finally perform the prediction. We study a number of dimensionality reduction methods and study how they affect the accuracy of prediction models.

Section 3.1 introduces and motivates the problem of quality estimation. Section 3.2 describes the proposed two-step training methodology. Section 3.3 describes the different dimensionality reduction methods, including the two novel methods proposed in this thesis. Section 3.4 describes the prediction models, and Section 3.5 describes the features used in the experimentation. The results of the experimentation are described in Section 3.6. Finally, the chapter ends with a summary in Section 3.7.

Chapter Outline

3.1 Introduction

It is clear that fully-automatic MT technology have greatly improved since its initial steps in the Georgetown experiment [IBM, 1954; Hutchins, 2005], however current state-of-the-art MT systems are still error-prone [Callison-Burch et al., 2012]. A desirable feature to improve the utility of MT technology is thus the capability of predicting at run-time the reliability, namely the quality, of the generated translations. This task, referred to as confidence or *quality estimation* (QE), is concerned about estimating MT output quality when reference translations are not available [Gandrabur and Foster, 2003; Blatz et al., 2004; Quirk, 2004; Gamon et al., 2005; Ueffing and Ney, 2007; Sanchis et al., 2007; Specia et al., 2009b]. Quality information is a crucial component in practical MT systems in the various possible scenarios involving the use of automatic translations. The following are some applications of quality information:

- Inform readers of sentences (or portions thereof) that are not reliable.
- Decide whether a given translation is good enough for publishing as is.
- Select which automatic translations should be supervised by the user (and used to re-train the MT model) in an active learning scenario.
- Select the best translation among options from multiple MT systems.

Quality information may be provided for each word [Gandrabur and Foster, 2003; Ueffing and Ney, 2007; Sanchis et al., 2007], sentence [Blatz et al., 2004; Quirk, 2004; Gamon et al., 2005; Specia et al., 2009b] or document [Soricut and Echihabi, 2010]. Here, we are specifically interested on the estimation of sentence level quality scores. However, under the assumption that the quality of a sentence is somehow related with the quality of its words, we also study how to utilize world-level indicators to estimate sentence-level quality.

Sentence-level QE is typically addressed as a regression problem [Quirk, 2004; Blatz et al., 2004; Specia et al., 2009b]. Given a translation (and potentially other additional sources of information), a set of features is extracted. Then, a predictor model is employed to compute a quality score from these features. This point of view provides a solid, well-know framework, within which accurate predictors can be derived. However, several particular problems arise when applying this approach to predict the quality of natural language sentences. For example, while the concept of translation quality is quite intuitive, the definition of automatically-computable features that reliably account for it has proven to be elusive. Thus, in practice, feature sets tend to contain a large number of noisy, collinear and ambiguous features.

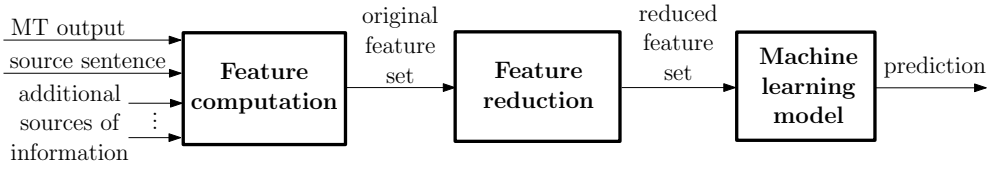


Figure 3.1: Dataflow of the proposed two-step training methodology.

We propose a training methodology for sentence-level QE specifically designed to address these challenges. We consider training as a two-step process. In an initial step, the system itself decides which are the actual latent variables that are relevant to perform the prediction. In other words, the QE system tries to extract, from the whole set of features provided, the latent variables that actually govern the quality of the translations. Specifically, we formalize this module as a *dimensionality reduction* (DR) problem. We then use the latent variables generated in the initial step to train the predictor model of our choice. Figure 3.1 shows a scheme of the proposed methodology to obtain a quality score from a given translation. Additionally, despite being tested in a QE task, the proposed two-step training and DR methods do not make particular assumptions about the features or the learning model. Thus, they constitute a general methodology that can be applied to a great variety of supervised learning tasks.

It should be noted that the proposed methodology involves a subtle focus shift in the way QE is conceived. Typical QE approaches use expert knowledge to try to define the features that are most informative to perform the prediction [Blatz et al., 2004]. We, on the contrary, plead for a simpler approach where every attribute or feature available is measured, and is the prediction system, rather than an expert, who is in charge of deciding which are the relevant variables. The potential advantages of the proposed methodology are two-fold. On the one hand, we expect to improve the accuracy of the prediction model since those noisy features that usually hinder model learning are filtered out. On the other hand, by reducing the number of features from which the actual prediction is made, we are reducing the response time of the QE system at both training- and prediction-time.

We propose two novel DR methods based on *partial least squares regression* (PLSR) [Wold, 1966]. The origin of PLSR lies in chemistry [Wold et al., 2001], but has been successfully applied in a wide range of applications such as statistical process control [Kresta et al., 1991], tumor classification [Nguyen and Rocke, 2002] or marketing [Fornell and Bookstein, 1982]. We develop both

a DR method that selects a subset of the original features, namely a feature selection method, and a method that projects the original data into a space of fewer dimensions, that is, a feature extraction method. Despite being usually more complex, feature extraction methods have a potential advantage over feature selection: they can generate new features that summarize the "information" contained in all original features. In contrast, the information contained in the features discarded by a feature selection method is inevitably lost.

Dimensionality reduction techniques have been previously applied in the QE literature. However, typically QE systems simply select a subset of features based on some kind of relevance measure. The proposed DR methods are compared to other methods previously used in the literature: methods based on statistical multivariate analysis such as *principal component analysis* (PCA) [Pearson, 1901] and PLSR regressors selection [Specia et al., 2009b; García-Martínez, 2012], and heuristic wrapper selection methods [Kohavi and John, 1997]. Moreover, we study how DR affect the performance of several prediction models.

The performance of each DR method was evaluated by the prediction accuracy of the models built with the corresponding reduced feature sets according to the proposed two-step training methodology. To assure a fair comparison between the different DR methods, identical pipelines were used to train the models. By providing a detailed description and a systematic evaluation of these DR methods, we give the reader various criteria for deciding which method to use for a given task.

3.2 Proposed Training Methodology for QE

Translation QE is usually formalized as a regression problem [Specia et al., 2009b] where we model the relationship between a dependent variable $y \in \mathbb{R}$ (the quality score of the translation), and a vector $\mathbf{x} \in \mathbb{R}^M$ of M explanatory variables $\mathbf{x}^T = \{x_1, \ldots, x_m, \ldots, x_M\}$[a] (the features that represent the translation sentence). Given a training set with N samples $\{y_n, \mathbf{x}_n\}_{n=1}^N$, our goal is to build a regression model $\mathbb{M}_{\boldsymbol{\theta}} : \mathbb{R}^M \rightarrow \mathbb{R}$, where $\boldsymbol{\theta}$ are its free parameters. The data set is usually represented in matrix form where $\mathbf{y}$ is a vector that contains the quality scores to be predicted, and $\mathbf{X}$ is a matrix where each row

[a] Given a vector $\mathbf{x}$, we use $\mathbf{x}^T$ to denote its transpose.

is the transposed feature vector of one training sample:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_N \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1m} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} & \cdots & x_{nM} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nm} & \cdots & x_{NM} \end{pmatrix}$$

To carry out the regression, the form of the model $\mathbb{M}_{\boldsymbol{\theta}}$ must be specified. Since we do not know how $\mathbf{y}$ and $\mathbf{X}$ actually relate, we can use different flexible models (further discussed in Section 3.4) whose free parameters $\boldsymbol{\theta}$ can be estimated to fit the data.

This approach to estimate QE systems requires the definition of a set of features $\mathbf{x}$ that aim to explain the quality of the generated translations. This process involves a great amount of effort in "feature engineering" to define the features from expert knowledge. Moreover, despite the great effort invested in the definition of features, there is still no general agreement on which are the features that best account for translation quality [Blatz et al., 2004; Callison-Burch et al., 2012].

Alternatively, we can devise a QE approach where the system automatically identifies which are the relevant features to estimate translation quality. In this scenario, no human effort is required for feature engineering; every potential feature could be measured and passed to the system. Since this approach obtains a reduced set of relevant features, it also tackles well-known learning problems related to the use of a large quantity of noisy features including the "curse of dimensionality" [Bellman, 1961], collinearity between features and feature ambiguity.

We divide the conventional QE regression problem ($\mathbb{R}^M \rightarrow \mathbb{R}$) into two independent sub-problems, see Figure 3.1. First, we implement a module that transforms a potentially highly-noisy M-dimensional set of features into a new R-dimensional set of features ($R < M$) suitable to train robust prediction models ($\mathbb{R}^M \rightarrow \mathbb{R}^R$). Then, we use this reduced set of features to train a model to predict the actual quality scores of the translations ($\mathbb{R}^R \rightarrow \mathbb{R}$). Note that this approach corresponds to the formalization of the ideas schematized in Figure 3.1. Next sections provide a description of different possible implementations for these two modules.

3.3 Dimensionality Reduction

3.3.1 Motivation

The proposed QE formalization assumes that translation quality is governed by a number of independent variables. Since these variables are usually unknown, in practice, we try to represent the prediction information contained in them by extracting a (possibly larger) set of features. This approach implies to consider translation quality as governed by more variables than it really is, which results in several learning problems due to the addition of irrelevant features, or the multicollinearity between them. However, provided the influence of this "extra" features is not too strong as to completely mask the original structure, we should be able to "filter" them out and recover the original variables or an equivalent set of them. DR methods aim at somehow strip off this redundant information, producing a more economic representation of the data.

DR can also be seen as a method to overcome the so-called "curse" of dimensionality. This term, coined in [Bellman, 1961], refers to the fact that, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables. Responsible for the "curse" of dimensionality is the fact that high-dimensional spaces are inherently sparse which is known as the empty space phenomenon [Scott and Thompson, 1983]. This is a difficult problem in model estimation since it states that the number of samples required for learning grows exponentially with the dimension of the space. DR technology address these problems by reducing the input dimension of the function to be estimated.

3.3.2 Dimensionality Reduction Problem and Approaches

The DR problem can be stated as follows: given a regression problem $\mathbb{R}^M \rightarrow \mathbb{R}$, we want to obtain an equivalent problem $\mathbb{R}^R \rightarrow \mathbb{R}$ where $R < M$. In other words, we want to obtain a low-dimensional, compact representation of the input data that still retains the information required to perform an accurate prediction. Formally, DR is defined by a projection Δ that transforms an M-dimensional space into an R-dimensional space:

$$\Delta : \mathbb{R}^M \rightarrow \mathbb{R}^R \tag{3.3.1}$$

The determination of the dimension R of this compact representation is central to the DR problem, because knowing it would eliminate the possibility of over- or under-fitting. All the DR methods studied in this chapter take this

intrinsic dimension as a parameter to be given by the user; a trial-and-error process is thus necessary to obtain a satisfactory value for it.

Next, we describe the different DR methods tested in the experimentation. For a more clear presentation, we distinguish between heuristic methods and methods derived from statistical multivariate analysis.

3.3.3 Heuristic Feature Selection Methods

We consider heuristic wrapper methods [Kohavi and John, 1997] to address the problem of feature selection. In the wrapper methodology, the learning model is considered a perfect black box. In its most general formulation, this methodology consists in using the prediction accuracy of a given learning model to assess the relative usefulness of subsets of features. In practice, the different wrapper methods are defined by the search strategy implemented to explore the space of possible subsets. An exhaustive search can conceivably be performed if the number of features is not too large. But, the problem is known to be NP-hard [Amaldi and Kann, 1998] and the search quickly becomes computationally intractable.

In the experimentation, we tested two search strategies that define two different heuristic feature selection methods: ranking of feature selection, and greedy forward selection. Since the computational complexity of these simple methods depends on the complexity of the chosen learning model, we use symbol $\zeta(N, M)$ to denote the time complexity to train the actual learning model with N samples of M-dimensional feature vectors.

Rank of Feature

Rank of feature selection (RFS) generates subsets of features by selecting the top-scoring features according to the prediction accuracy of a QE system trained solely with that feature. RFS is typically used as a baseline selection mechanism because of its simplicity, scalability and (somewhat) good empirical success [Guyon and Elisseeff, 2003]. The computational complexity of RFS to generate the first reduced feature set is given by $O(M \cdot \zeta(N, 1))$; once the scores for the features are computed, we can generate reduced groups of different sizes with no further calculations. For example, the complexity of RFS if we use a linear model[b] is in $O(M \cdot N)$ given that $\zeta(N, 1)$ is proportional to N.

Since RFS selects the features according to their individual prediction accuracy, we expect to obtain subsets of features that also provide good prediction

[b] This particular setup can be considered as a lower bound of the complexity of RFS.

accuracy. However, RFS does not take into account the correlations that may exist between the different features, thus, these subsets will probably contain a large number of redundant features.

Greedy Forward

Greedy forward selection (GFS) [Kohavi and John, 1997; Avramidis, 2012] incrementally creates subsets of features by selecting at each iteration the feature that, when added to the current set, yields the learned model that performs best. In contrast to RFS, GFS recomputes the importance of each feature at each step having into account the current subset of features. Thus, the computational complexity of GFS to compute a reduced set of size R is in $O(\sum_{r=1}^{R}\sum_{m=1}^{M-r+1}\zeta(N,m))$ that can be upper bounded by $O(R\cdot M\cdot\zeta(N,R))$. For example, if we use a linear model the temporal complexity of GFS is in $O(R^2\cdot M\cdot N)$ given that $\zeta(N,R)\propto N\cdot R$.

Since GFS selects at each step the feature that improves most the QE model performance, we expect to obtain subsets with lower redundancy in comparison to RFS. However, it requires to re-compute the contribution of each feature to the QE model at each step, $O(\zeta(N,R))$, which penalizes GFS complexity.

3.3.4 DR Methods Based on Statistical Multivariate Analysis

Statistical multivariate analysis is a generic term for any statistical technique concerned with analyzing data in high dimensions [Anderson, 1958]. In particular, we focus on statistical techniques to partition the variability of the data into components attributable to different sources of variation. In this work, we consider two of these techniques: principal component analysis (PCA), and partial least squares regression (PLSR). Given a number of dimensions R, both PCA and PLSR can be used to compute a transformation of the original data space into an orthogonal R-dimensional space. However, they differ in the criteria followed to compute this transformation.

The main advantage of these methods stems in the orthogonality of the output space; which means that the transformed features will be linearly independent by construction. Therefore, using these transformations we obtain reduced feature sets with almost no redundant information. Moreover, statistical multivariate methods are mathematically well-founded and independent of the chosen learning model. However, these methods also have an obvious drawback, i.e. new features are computed as a linear combination of all original features which makes it often difficult to interpret them.

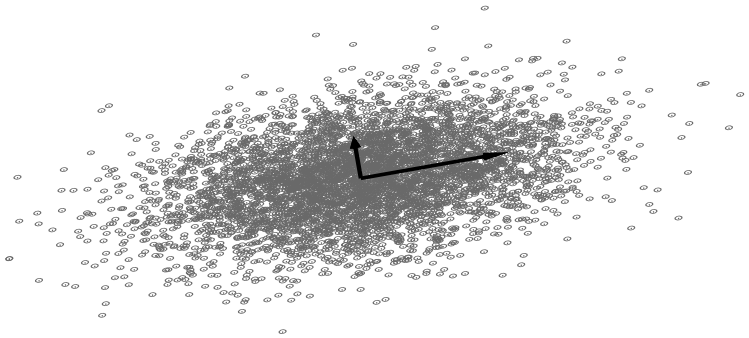


Figure 3.2: PCA example for a 2-dimensional gaussian distribution. The vectors represent the two principal directions of variation (eigenvectors) of the data.

Principal Component Analysis

Principal component analysis (PCA) defines a transformation of the original data into a new space of features, known as principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint of being uncorrelated with the preceding components. Therefore, each of these principal components represent one of the individual latent factors that actually govern the variability of the data, as exemplified in Figure 3.2.

Given a matrix $\mathbf{X}$ whose rows represent the N samples and each column represents one of the M features, PCA is formalized by the following decomposition:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T \tag{3.3.2}$$

where $\mathbf{P}$ is the space transformation matrix that contains the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$, and the rows of $\mathbf{T}$ represent the principal components of each training sample. The nonlinear iterative partial least squares (NIPALS) algorithm [Wold, 1966] is commonly used to obtain the eigenvectors.

Given that the eigenvectors in $\mathbf{P}$ are unitary and orthogonal ($\mathbf{P}^T\mathbf{P} = \mathbf{I}$), we can multiply both sides of Equation (3.3.2) by $\mathbf{P}$ to obtain the principal

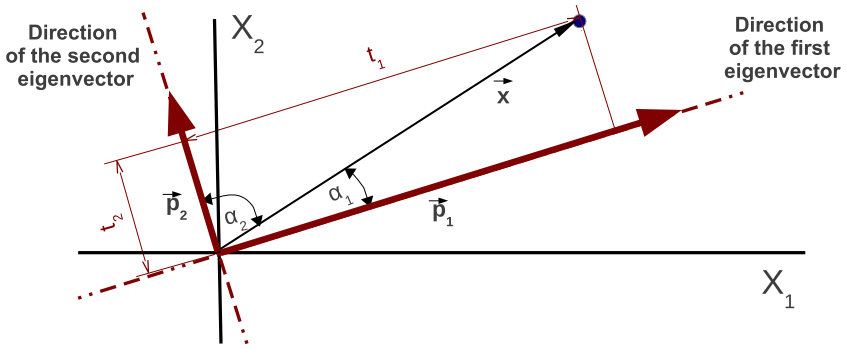


Figure 3.3: Example of the principal component values (t_1, t_2) for a data point $\mathbf{x}$ in Figure 3.2. Values t_1, t_2 are computed by projecting $\mathbf{x}$ over the corresponding eigenvectors $(\vec{p}_1, \vec{p}_2)$.

components $\mathbf{T}$ of the data:

$$\mathbf{XP} = \mathbf{T} \tag{3.3.3}$$

Figure 3.3 shows a graphical example of the computation of two principal components $\mathbf{t} = (t_1, t_2)$ for a single data point $\mathbf{x}$. Each principal component t_r is computed by projecting $\mathbf{x}$ over the corresponding unitary eigenvector $\mathbf{p}_r$. Specifically, $t_r = \mathbf{x} \cdot \mathbf{p}_r = ||\mathbf{x}|| \cdot ||\mathbf{p}_r|| \cdot \cos(\alpha_r) = ||\mathbf{x}|| \cdot \cos(\alpha_r)$, where α_r is the angle between $\mathbf{x}$ and $\mathbf{p}_r$.

PCA Projection

The principal components are linearly independent, and each of them accounts for the maximum variability in $\mathbf{X}$ not explained by previous components. Taking this into account, we can obtain reduced feature sets by selecting the first R principal components, that is, by selecting the first R columns in matrix $\mathbf{T}$. Since each of these components is a linear combination of the original features, this is a feature extraction method. In the experiments, we refer to this particular DR method as PCA-P.

The complexity of PCA-P to compute a reduced set of size R is given by the complexity of the NIPALS algorithm: $O(R \cdot M \cdot N)$. Note that in contrast to the previously presented heuristic methods, the cost of PCA-P does not depend on the complexity of the chosen learning model.

Partial Least Squares Regression

PCA generates sets of orthogonal features where each feature explains the variability of the data $\mathbf{X}$ in one principal direction. However, this transformation ignores the scores $\mathbf{y}$ to be predicted. Thus, although the features generated by PCA-P contain almost no redundancy, they do not necessarily have to be the best set of features to perform the prediction. Partial least squares regression (PLSR) [Wold, 1966] is an alternative to PCA that takes into account $\mathbf{y}$ when computing the transformation of $\mathbf{X}$. Specifically, PLSR computes an ordered set of latent variables such that each of them account for the maximum co-variability between $\mathbf{X}$ and $\mathbf{y}$ under the constraint of being uncorrelated with previous latent variables. In other words, latent variables are extracted so they better explain the values to be predicted. As a result, the number of latent variables required to reach a certain prediction accuracy is usually much lower than the number of principal components required to obtain the same accuracy.

Formally, PLSR builds the following model where $\mathbf{b}$ is a vector of regressor coefficients, and $\mathbf{f}$ is a vector of zero-centered Gaussian errors:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{f} \tag{3.3.4}$$

Even though this is a linear regression model the estimation of the regression coefficients $\mathbf{b}$ for PLSR is different from the conventional least squares regression, see Section 3.4.1. The intuitive idea of PLSR is to describe $\mathbf{y}$ as well as possible, hence to make $||\mathbf{f}||$ as small as possible, and, at the same time, take advantage of the relation between $\mathbf{X}$ and $\mathbf{y}$. To do that, PLSR defines two independent PCA-like transformations $\mathbf{P}$ and $\mathbf{q}$ (for $\mathbf{X}$ and $\mathbf{y}$ respectively) with $\mathbf{E}$ and $\mathbf{f}$ being the corresponding residual errors, and a linear relation $\mathbf{R}$ linking both blocks:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \qquad \mathbf{y} = \mathbf{U}\mathbf{q}^T + \mathbf{f} \tag{3.3.5}$$

$$\mathbf{U} = \mathbf{T}\mathbf{R} \tag{3.3.6}$$

where matrices $\mathbf{T}$ and $\mathbf{U}$ are the projections from $\mathbf{X}$ and $\mathbf{y}$ respectively. Specifically, each of the columns of the $\mathbf{T}$ matrix represents one of the latent variables of $\mathbf{X}$. The NIPALS algorithm [Wold, 1966] is also used to solve this optimization problem. In this case, $\mathbf{b}$ is estimated as:

$$\mathbf{b} = \mathbf{R}\mathbf{q}^T \quad \text{where} \quad \mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1} \tag{3.3.7}$$

where $\mathbf{W}$ is an internal weight matrix used by the algorithm that accounts for the correlation between $\mathbf{X}$ and $\mathbf{U}$. An exhaustive description of the NIPALS algorithm for PLSR can be found in [Geladi and Kowalski, 1986].

Since PLSR is a much more sophisticated model than PCA, different PLSR-based DR methods can be derived. In addition to the regressors-based selection method previously described in [Specia et al., 2009b], we propose one new feature selection method, variance importance in projection, and one new feature extraction method, PLSR projection. Similarly to PCA-P, the computational complexity of these three PLSR-based DR methods is also given by the complexity of the NIPALS algorithm, $O(R \cdot M \cdot N)$.

Feature Importance in Regression

Let us consider the predictions of the PLSR linear model (Equation (3.3.4)):

$$\hat{y} = b_1 x_1 + \cdots + b_m x_m + \cdots + b_M x_M \tag{3.3.8}$$

Regressor scores b_m denote the expected value increment of the predicted quality score $\hat{y}$ by unitary increment of feature x_m, i.e., they denote the importance of each feature in the regression. However, due to the usually different scale of the features, these values cannot be directly compared; first, data need to be standardized by subtracting the feature mean from the raw data values and dividing the difference by the standard deviation. Standardized features become dimensionless, and then regressors are directly comparable. We then can create a reduced set of features by selecting them in descending regressor absolute value ($\mathbf{b}$ in Equation (3.3.4)) [Specia et al., 2009b]. We use FIR (*Feature Importance in Regression*) to denote this method in the experiments.

Variance Importance in Projection

Given the weight matrix $\mathbf{W}$, we can compute the importance of each original feature has in the computation of the projection. This value can be computed by the *variance importance in projection* (VIP) [Chong and Jun, 2005] of each feature. VIP is a score that evaluates the importance of each feature to find the R latent variables. Therefore, similarly as done for RFS in Section 3.3.3, we propose to select subsets of top-scoring features according to their VIP score. The VIP score for the m^{th} feature is computed as:

$$\text{VIP}_m = \sqrt{\frac{M \cdot \sum_{r=1}^{R} \left(\frac{w_{mr}}{||\boldsymbol{w}_r||}\right)^2 \cdot \text{ESS}_r}{\sum_{r=1}^{R} \text{ESS}_r}} \tag{3.3.9}$$

where M is the number of original features, $\text{ESS}_r = b_r^2 \mathbf{t}_r^T \mathbf{t}_r$ is the square of the contribution of the r^{th} latent variable to the score predicted by the PLSR model, $\mathbf{t}_r$ is the r^{th} column of matrix $\mathbf{T}$, b_r is the r^{th} regressor coefficient in

$\mathbf{b}$, and $\frac{w_{mr}}{||\boldsymbol{w}_r||}$ is the normalized value of weight w_{mr}.

PLSR Projection

The latent variables are linearly independent, and each of them accounts for the maximum co-variability between $\mathbf{X}$ and $\mathbf{y}$ not explained by previous latent variables, thus we propose to obtain a reduced feature set by extracting the first r latent variables, i.e. the first r columns in matrix $\mathbf{T}$. In contrast to PCA, the latent variables computed by PLSR take into account the relation between the features $\mathbf{X}$ and the quality scores $\mathbf{y}$. Therefore, in addition of being linearly independent, we expect the latent variables to attain more predictive potential than the equivalent number of principal components. This feature extraction method is labeled PLS-P in the experiments.

3.4 Machine Learning Models

Now, we describe the particular models implemented to solve the remaining regression problem $\mathbb{R}^R \rightarrow \mathbb{R}$ where we actually predict a quality score $y \in \mathbb{R}$ from a reduced set of features $\mathbf{x} \in \mathbb{R}^R$ generated by the previous DR module. Next, we describe different flexible learning models $\mathbb{M}_{\boldsymbol{\theta}}$ whose free parameters $\boldsymbol{\theta}$ can be estimated so that the model best fits the data. In the experiments, we used the WEKA toolkit [Hall et al., 2009] to build the different models.

3.4.1 Linear Regression

Linear regression assumes a linear relationship between the prediction value y_i and the vector of features $\mathbf{x}_i$. This relationship is modeled by a vector of weights $\boldsymbol{\theta}^T = (\theta_1, \ldots, \theta_r, \ldots, \theta_R)$[c]. Formally, linear regression models take the form of a set of equations:

$$y_i = \theta_1 x_{i1} + \cdots + \theta_r x_{ir} + f_i, \quad i = 1, \ldots, n \tag{3.4.1}$$

where n is the number of training samples, r is the number of features, and f_i are zero-centered Gaussian error variables. Often all equations are stacked together and written in matrix form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{f} \tag{3.4.2}$$

The most common technique to estimate the free parameters $\boldsymbol{\theta}$ of linear models is known as least squares estimation. This method minimizes the sum

[c]We change the notation with respect to the formulation of the PLSR model in Equation (3.3.4) because each model follows a different process to estimate its free parameters.

of squared errors, and leads to a closed-form expression for the optimum values of $\boldsymbol{\theta}$:

$$\widehat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.4.3}$$

Additionally, different regularization techniques are usually implemented to prevent ill-posed learning problems when multicollinearity is present. Regularization techniques deliberately introduce bias into the estimation of $\widehat{\boldsymbol{\theta}}$ to penalize complex models. In the experiments, we used ridge and LASSO regression [Tibshirani, 1996]. Both methods constraint the norm of the parameter vector (L^2-norm ridge and L^1-norm LASSO) to be lower than a given value γ.

3.4.2 Support Vector Machines

In practice, few natural phenomena exhibit a linear relationship between their explanatory variables $\mathbf{x}$ and the corresponding dependent variable y. Thus, linear regression cannot adequately describe such nonlinear phenomena.

Support vector machines [Cortes and Vapnik, 1995] (SVMs) are a class of machine learning models that, as linear regression, assume a linear relationship between $\mathbf{X}$ and $\mathbf{y}$. However, prior to any calculation, SVMs project the data into an alternative space. This projection, defined by a kernel function $\varphi(\mathbf{x})$, may be nonlinear; thus, though a linear relationship is learned in the projected feature space, this relationship may be nonlinear in the original input space. Choice of the kernel determines whether the resulting SVM is a polynomial regressor, a two-layer neural network, a radial basis function machine, or some other learning machine [Abe, 2010].

The linear relationship is estimated as a regularized (L^2-norm) optimization problem. In contrast to linear regression, the SVM model depends only on a subset of the training data, because the cost function for building the model does not care about those training samples that already lie within a given margin. There exist several specialized algorithms for quickly solving the quadratic programming problem that arises. For example, sequential minimal optimization [Platt, 1999] breaks the problem down into 2-dimensional sub-problems that can be solved analytically.

Preliminary experiments studying different kernels showed that radial basis kernel obtained among the best results and additionally was easier to train than other kernels such as polynomial kernels. Therefore, in the experimentation we used SVMs with a radial basis kernel.

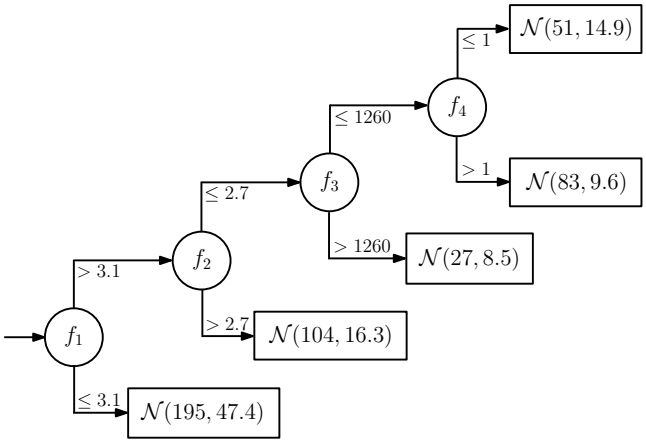


Figure 3.4: Example of a regression tree. It uses four feature comparisons to partition the data-space, and gaussian normal distributions to model the data on each of the five partitions.

3.4.3 Regression Trees

Typical regression models, such as linear regression or SVMs, are global. In other words, there is a single predictive formula holding over the entire data-space. When the data has lots of features which interact in complicated, nonlinear ways, assembling a single global model can become a very difficult problem. An alternative approach is to recursively partition the data-space into smaller regions, until they are simple enough to fit elemental models to them.

Regression trees use a tree structure to represent such a recursive partition. Each of the terminal nodes of the tree represents a region of the partition, and has attached to it a simple model which applies in that region only. We start at the root node of the tree, and ask a sequence of questions about the features. The interior nodes are labeled with questions, and the edges between them are labeled with the answers. Typically, each question refers to only a single feature, and has a yes or no answer, e.g., "Is Horsepower > 50?" or "Is GraduateStudent == FALSE?". Features can be of different types (continuous, discrete, categorical, etc), and more-than-binary questions can be done, but these can always be accommodated as a larger binary tree. Figure 3.4 shows an example of a regression tree using gaussian normal distributions to model the data on each partition.

Once we fix the tree structure, local models are completely determined and easy to find, so all the effort should go into finding a good tree structure, which is to say into finding a good partitioning of the data-space. In our experiments,

we specifically use M5 regression trees [Quinlan, 1992] because one of the best submissions [Soricut et al., 2012] to the 2012 QE task [Callison-Burch et al., 2012] used such model.

3.5 Features

We computed 480 features to represent each automatic translation. Most of these features have been described in previous works for translation QE [Blatz et al., 2004; Ueffing and Ney, 2007; Sanchis et al., 2007; Specia et al., 2009a; Callison-Burch et al., 2012]. Some features are highly-correlated, for example, we considered both the translation probability and the perplexity given by a language model. As described in Section 3.3.1, working with such redundant features involves several learning issues. However, these inherent problems make translation QE a task where DR techniques may lead to important improvements in prediction accuracy.

Attending to [Specia et al., 2009b], the extracted features could be classified into *black-box* features and *glass-box* features. On the one hand, black-box features (B) consider the QE system as a black-box whose internals are not accessible. Hence, these features are computed directly from the input sentence and the automatic translation, being independent of the MT system that generates the translation. On the other hand, glass-box (G) features may also depend on some aspect of the translation process. In other words, glass-box features are specific for the chosen MT system.

Additionally, we also distinguish between sentence-based and subsequence-based features. Sentence-based features consider the translated sentence as an atomic unit and are computed from the translation as a whole. In contrast, subsequence-based features consider the translation as a sequence, and describe the quality of the whole translation as a combination of the feature scores of the subsequences contained in it. The key idea of subsequence-based features is that the quality of a translation is related to the quality of the elements that conform it. Therefore, an adequate combination of subsequence-level features may become a good estimator of sentence-level quality.

3.5.1 Data

In the experiments, we estimated the quality of the translations of the English–Spanish (En–Es) news evaluation data used in the shared QE task[d] featured at the 2012 workshop on statistical MT [Callison-Burch et al., 2012]. Those

[d] http://statmt.org/wmt12/quality-estimation-task.html

	Training	Test
Sentences	1832	422
Tokens	49706	10095
Avg. sentence Length	18.6	24.0
Vocabulary	12284	3601

Table 3.1: Main figures of the training and test automatic Spanish translations for which we predicted translation quality.

translations were generated by a phrase-based MT system [Koehn et al., 2007] trained on the Europarl [Koehn and Monz, 2006] and the News Commentaries [Callison-Burch et al., 2007] corpora as provided for the shared translation task[e]. Evaluation data contains 1832 translations for training and 422 translations for test. Table 3.1 displays the main figures of the training and test corpora. Each translation was manually scored by several professional translators in terms of post-editing effort according to the following scheme:

[1] The translation is incomprehensible. It must be translated from scratch.

[2] About 50%–70% of the translation needs to be edited to be publishable.

[3] About 25%–50% of the translation needs to be edited.

[4] About 10%–25% of the translation needs to be edited.

[5] The translation is clear and intelligible. It requires little to no editing.

The final quality score of each translation (a real number in the range $[1, 5]$) is the average of the scores given by the different experts. Additionally, for each translation the corresponding source sentence, and decoding information (decoding graph and 1000-best alternative translations) are available. We used these and the training data of the shared translation task to compute the 497 features that describe each translation in the training and test corpora.

3.5.2 Sentence-Based Features

Sentence Length *(*B*, 3 features, [Blatz et al., 2004])*

We used three features based on the length of the sentences:

- Source sentence length.

[e] http://statmt.org/wmt12/translation-task.html

- Translation length.
- Source / translation length ratio.

These features are trivial, but could be indicators of the intrinsic difficulty of the source sentence and of the potential mismatch between source sentence and translation.

Language Model *(*B*, 30 features, [Blatz et al., 2004])*

Using the training data of the shared translation task, we built n-gram language models with n ranging from one to five for both the source and target languages. Then, for each source sentence and translation, we extracted the following three features:

- Log-probability of the sentence.
- Log-probability divided by sentence length.
- Perplexity of the sentence.

The language models with Kneser-Ney discounting as backoff were built using the SRI Toolkit. Finally, a total of 30 features based on language models were extracted (three features $\times$ five n-gram sizes $\times$ {source,translation}).

N-best Translations *(*G*, 3 features, [Blatz et al., 2004])*

The following features were extracted from the 1000-best translations provided for each source sentence.

- Average length of the translations in the N-best list.
- Vocabulary size of the N-best list divided by the average length.
- Vocabulary size divided by source sentence length.

Note that the last feature is a kind of average "fertility" of the words in the current source sentence.

N-best Language Model *(*G*, 30 features, [Blatz et al., 2004])*

In addition to the previous features that describe the N-best list as a whole, we computed various features from language models trained on the N-best list of translations. For each translation, we considered both language models trained using the particular N-best alternative translations of the source sentence, and using the full N-best list containing the N-best translations of all source sentences. Specifically, for each translation we computed the following features:

- Log-probability of the translation.
- Log-probability divided by translation length.
- Perplexity of the translation.

We used n-gram language models with n ranging from one to five with Kneser-Ney discounting for a total of 30 features (three features $\times$ five n-gram sizes $\times$ {particular N-best translations, full N-best list}).

These features try to capture the homogeneity of alternative translations of the same source sentence. This can be seen as an indicator of how confident the MT system is in the generated translation (the first translation of the N-best list); the higher the homogeneity the higher the confidence of the system.

Decoding Process *(G, 19 features, [Blatz et al., 2004; Specia et al., 2009a])*

We computed several features related to the decoding process that generated each translation:

- Percentage of "dead" nodes[f] in the search graph (one feature).
- Number of source phrases of sizes one to six explored through decoding (six features).
- Number of alternative translations considered during decoding for source phrases of sizes one to six (six features).
- Average size of the translations considered during decoding for source phrases of sizes one to six (six features).

Similarly as the N-best language model features, the objective of the decoding features is to estimate how difficult is for the MT system to translate the source sentence.

3.5.3 Subsequence-Based Features

In addition to the sentence-level features described above, a number of different features have been proposed in the literature at other granularity levels. Particularly, works on word-level QE have proposed a number of features that have shown good accuracy in predicting the quality of individual words [Blatz et al., 2004; Ueffing and Ney, 2005, 2007; Sanchis et al., 2007]. Thus, under the intuition that the quality of the individual words in a translation is somehow related with the quality of the full translation, we defined new

[f] Nodes no further expanded in the search graph.

sentence-level features as a combination of the quality scores of the words in each sentence. Moreover, we extended the concept of word-level feature to that of subsequence-level feature. In other words, instead of computing the quality for each individual word in a sentence, we computed features for each sequence of consecutive words in the sentence. Since the number of subsequences in a sentence is quadratic in the number of words in the sentence, we limit the size of the subsequences up to four words. Note that word features are a special case of subsequence features for subsequences of size one.

Once the subsequence features were computed, we have to combine them somehow in order to obtain the sentence-level feature required to predict the quality of each translation. Specifically, we represent each subsequence feature by five sentence-level indicators: the average value of the subsequence scores in the sentence and the percentage of the scores belonging to each frequency quartile. We used the training data of the shared translation task to compute these frequency quartiles. Each method represent a different approach to summarize the subsequence scores. The average value is a rough indicator that measures the "middle" value of them, while the quartile percentages are more fine-grained indicators that denote how spread out the scores are.

Source Subsequence Frequency (B, *24 features, [Blatz et al., 2004]*)

We compute the number of times each of the subsequences in the source sentence appears in the training data of the shared translation task. The frequencies for each subsequence size (one to four) are represented by five sentence-level indicators as described above, and an additional indicator denoting the number of subsequences with zero frequency. Finally, we compute a total of 24 features (four sizes $\times$ six sentence-level indicators).

These features are meant to reflect how common the subsequences in a given source sentence are on average. The intuition behind them is that if a large percentage of the subsequences in the source sentence have often been seen in the training corpus, then the translations produced for the sentence may be more accurate.

Word-to-Word Lexicon Probabilities *(B, 6 features, [Blatz et al., 2004; Ueffing and Ney, 2005])*

In [Brown et al., 1993], five SMT models were presented, from Model 1 to Model 5. While these models have been largely outperformed by modern translation approaches, we can still use them to examine the resulting translations, particularly the simpler Model 1. Model 1 uses what is known as a bag-of-

words translation model, meaning that its calculations are not tied to any specific alignment structure (apart from the basic one-to-many source-target correspondence assumed by all five models in [Brown et al., 1993]). Rather, for each source sentence and translation, we find the sum of probabilities of all possible alignments. This captures a sort of topic or semantic coherence in translations.

As described in [Brown et al., 1993], Model 1 computes the probability of a translation $\mathbf{e} = \mathrm{e}_1 \dots \mathrm{e}_i \dots \mathrm{e}_{|\mathbf{e}|}$ given a source sentence $\mathbf{f} = \mathrm{f}_0 \dots \mathrm{f}_j \dots \mathrm{f}_{|\mathbf{f}|}$ with the formula:

$$\mathrm{P}(\mathbf{e} \mid \mathbf{f}) = \frac{\epsilon}{(|\mathbf{f}|+1)^{|\mathbf{e}|}} \prod_{i=1}^{|\mathbf{e}|} \sum_{j=0}^{|\mathbf{f}|} \mathrm{P}(\mathrm{e}_i \mid \mathrm{f}_j) \tag{3.5.1}$$

where f_0 is the "empty" or "null" word, introduced to capture a target word that corresponds to no actual source word, ϵ is the probability of a translation of this particular size given the source sentence, and $\mathrm{P}(\mathrm{e}_i \mid \mathrm{f}_j)$ is the word-to-word lexicon, namely the probability for target word e_i of being the translation of source word f_j.

As we have said, Model 1 has been outperformed by most recent MT models, however it has shown very good performance when used to compute the quality of individual words. In this case, the quality of a given word e can be estimated by its contribution to the total Model 1 probability of the translation it belongs to:

$$\mathrm{P}(\mathrm{e} \mid \mathbf{f}) = \frac{1}{|\mathbf{f}|+1} \sum_{j=0}^{|\mathbf{f}|} \mathrm{P}(\mathrm{e} \mid \mathrm{f}_j) \tag{3.5.2}$$

This average probability has been used as word-level feature to estimate the quality of translated words [Blatz et al., 2004; Ueffing and Ney, 2004]. However, Ueffing and Ney [2005] proposed to substitute the average by the maximal lexicon probability since they discovered that the average was dominated by this maximum. The quality of a word is then given by:

$$\Upsilon(\mathrm{e} \mid \mathbf{f}) = \max_{0 \le j \le |\mathbf{f}|} \mathrm{P}(\mathrm{e} \mid \mathrm{f}_j) \tag{3.5.3}$$

This word-level feature has been successfully used since then in multiple works on QE showing good prediction accuracy [Ueffing and Ney, 2005, 2007; Sanchis et al., 2007].

For each word in a given translation, we compute a word-level feature according to Equation (3.5.3). These scores are then combined into the five

sentence-level indicators previously described. Additionally, we compute one more sentence-level indicator denoting the number of words in the translation with zero[g] probability according to Equation (3.5.3).

Translation Options *(B, 5 features, [Specia et al., 2009a])*

Additionally, we also use a word-to-word lexicon probability to compute the number of possible alternative translations for each word in the source sentence. This feature estimates the ambiguity that exist in the translation of each source word, and thus denotes how difficult is the source sentence to translate. Finally, we represent this word feature with the five sentence-level indicators described at the beginning of the section.

***N*-best Posterior Probabilities** *(B, 288 features, [Ueffing et al., 2003; Sanchis et al., 2007])*

We extract a set of word-level features based on posterior probabilities computed over N-best lists [Ueffing et al., 2003; Blatz et al., 2004; Ueffing and Ney, 2007; Sanchis et al., 2007]. Consider a target word e_i belonging to a translation $\mathbf{e} = \mathrm{e}_1 \dots \mathrm{e}_i \dots \mathrm{e}_{|\mathbf{e}|}$ generated from a source sentence $\mathbf{f}$. Let $\mathcal{L}(\mathbf{f})$ be the ordered list of N-best alternative translations of the source sentence ($|\mathcal{L}(\mathbf{f})| = N$). We then determine those sentences in the N-best list that "contain" e_i under certain conditions that are to be explained later in this section. Let the set of those translations be $\mathcal{S}(\mathrm{e}_i) \subseteq \mathcal{L}(\mathbf{f})$. We compute four different features that are calculated based on relative frequencies, rank weighted frequencies, word posterior probabilities or weighted posterior probabilities.

The feature based on relative frequencies is computed as:

$$\frac{|\mathcal{S}(\mathrm{e}_i)|}{N} \tag{3.5.4}$$

The feature based on rank weighted frequencies is given by:

$$\frac{2}{N(N+1)} \sum_{\mathbf{e}' \in \mathcal{S}(\mathrm{e}_i)} \big(N + 1 - \mathrm{rank}(\mathbf{e}', \mathcal{L}(\mathbf{f}))\big) \tag{3.5.5}$$

where function $\mathrm{rank}(\mathbf{e}', \mathcal{L}(\mathbf{f}))$ returns the position in which translation $\mathbf{e}'$ appears in the list of N-best translations. We sum up the inverted ranks $N + 1 - \mathrm{rank}(\mathbf{e}', \mathcal{L}(\mathbf{f}))$, because we want an occurrence of word e_i in a translation near the top of the N-best list to score better than one in the lower ranks. This value is normalized by the sum of all ranks in the list.

[g] We consider as zero any probability below 10^{-7}.

In the case of statistical translation models, we can also compute features based on the posterior probabilities of the translations in the N-best list.

Let $\mathrm{P}(\mathbf{e} \mid \mathbf{f})$ the probability assigned by an SMT model to translation $\mathbf{e}$ given source sentence $\mathbf{f}$. The feature based on posterior probabilities is then calculated as the normalized sum of probabilities of all translations $\mathcal{S}(\mathrm{e}_i)$ that contain the word e_i:

$$\frac{1}{\sum_{\mathbf{e}'' \in \mathcal{L}(\mathbf{f})} \mathrm{P}(\mathbf{e}'' \mid \mathbf{f})} \sum_{\mathbf{e}' \in \mathcal{S}(\mathrm{e}_i)} \mathrm{P}(\mathbf{e}' \mid \mathbf{f}) \tag{3.5.6}$$

Note that the probability mass of all possible translations of a source sentence ($\sum_{\mathbf{e}''} \mathrm{P}(\mathbf{e}'' \mid \mathbf{f})$) should be equal to one and hence not necessary in the formulation. However, N-best lists represent only a relatively small portion of the space of possible translations. Thus, the need for the normalization term (the actual probability mass in the list of N-best translations $\mathcal{L}(\mathbf{f})$) in Equation (3.5.6).

As an alternative, we can compute a feature based on weighted posterior probabilities. The intuitive idea is that the probability mass of a translation should be equally distributed between its words. Therefore, to compute this feature, the posterior probability $\mathrm{P}(\mathbf{e}'' \mid \mathbf{f})$ of each translation $\mathbf{e}'' \in \mathcal{L}(\mathbf{f})$ has to be multiplied by the number of times $\#_{\mathrm{e}_i}(\mathbf{e}'')$ the word e_i appears in $\mathbf{e}''$ and divided by the total number of words in $\mathbf{e}''$:

$$\frac{1}{\sum_{\mathbf{e}'' \in \mathcal{L}(\mathbf{f})} \frac{\#_{\mathrm{e}_i}(\mathbf{e}'')\mathrm{P}(\mathbf{e}'' \mid \mathbf{f})}{|\,\mathbf{e}''\,|}} \sum_{\mathbf{e}' \in \mathcal{S}(\mathrm{e}_i)} \frac{\#_{\mathrm{e}_i}(\mathbf{e}'')\mathrm{P}(\mathbf{e}' \mid \mathbf{f})}{|\,\mathbf{e}'\,|} \tag{3.5.7}$$

Similarly as for Equation (3.5.6), if we had the full probability distribution $\mathrm{P}(\mathbf{e}'' \mid \mathbf{f})$ instead of an N-best list the denominator could be simplified. In that case, the denominator would be $\sum_{\mathbf{e}'' \in \mathcal{L}(\mathbf{f})} \frac{\#_{\mathrm{e}_i}(\mathbf{e}'')}{|\,\mathbf{e}''\,|}$.

The four weighting schemes described above are computed from a subset $\mathcal{S}(\mathrm{e}_i) \subseteq \mathcal{L}(\mathbf{f})$ of the translations in the N-best list. Specifically, the translations in $\mathcal{S}(\mathrm{e}_i)$ are those that "contain" the word e_i. Next, we describe the three different criteria implemented to compute $\mathcal{S}(\mathrm{e}_i)$:

- $\mathcal{S}(\mathrm{e}_i) = \{\mathbf{e}' \in \mathcal{L}(\mathbf{f}) \mid \mathbf{a} = \mathrm{Levenshtein}(\mathbf{e}', \mathbf{e}) \wedge \mathrm{e}'_{a_i} = \mathrm{e}_i\}$

  The translation $\mathbf{e}$ and the translations $\mathbf{e}' \in \mathcal{L}(\mathbf{f})$ in the N-best list may exhibit different word orders, thus we first align each $\mathbf{e}'$ to $\mathbf{e}$ to estimate a correspondence between the words of both translations. We do that using a Levenshtein alignment [Levenshtein, 1966] $\mathbf{a} = a_1 \ldots a_i \ldots a_{|\,\mathbf{e}'\,|}$. Then, we select those translations $\mathbf{e}'$ for which the word in position a_i, i.e. the word aligned to the position i in $\mathbf{e}$, is equal to the word e_i.

- $\mathcal{S}(\mathrm{e}_i) = \{\mathbf{e}' \in \mathcal{L}(\mathbf{f}) \mid \mathrm{e}'_i = \mathrm{e}_i\}$

  A second option is to consider that all translations share the same word order. Under this consideration, we select all sentences $\mathbf{e}'$ that contain the word e_i in exactly the target position i.

- $\mathcal{S}(\mathrm{e}_i) = \{\mathbf{e}' \in \mathcal{L}(\mathbf{f}) \mid \exists i' : \mathrm{e}'_{i'} = \mathrm{e}_i\}$

  The previous criteria are too strict. Often, the same target word occurs in several translations, but in different positions due to reordering of words, insertions and deletions. Here, we relax the previous criteria and select all sentences that contain the target word e_i, disregarding its position.

By combining a weighting scheme (relative frequency, rank weighted frequency, posterior probability, and weighted posterior probability) and a implementation of $\mathcal{S}(\mathrm{e}_i)$, we can compute a total of 12 word-level features.

Additionally, we extend the definition of these word-features to operate at sequence level. The intuitive idea of this extension is that the quality of individual words is influenced by the words in its context. Therefore, we operate with sequences to take into account the context of a word in the computation of its quality. To do that, we can simply transform each sentence from a sequence of words to the corresponding sequence of sequences. For example, given the sentence "we are faced with enormous challenges", we obtain the following sentences depending on the selected subsequence size (one to four):

- "we are faced with enormous challenges".
- "we_are are_faced faced_with with_enormous enormous_challenges".
- "we_are_faced are_faced_with faced_with_enormous with_enormous_challenges".
- "we_are_faced_with are_faced_with_enormous faced_with_enormous_challenges".

Given a subsequence size, we first generate the corresponding *sequentialized* versions of $\mathbf{e}$ and $\mathbf{e}' \in \mathcal{L}(\mathbf{f})$. Then, we simply compute the 12 features described above for the newly generated sentences.

Summing up, we can compute 48 subsequence-level features (four sizes $\times$ four weighting schemes $\times$ three implementations of $\mathcal{S}(\mathbf{f})$). For each of these subsequence-level features, we then compute the five sentence-level indicators described at the beginning of this section, accounting for 240 sentence-level indicators. Additionally, we also compute the number of subsequences with zero ($< 10^{-7}$) posterior probability, accounting for 48 additional features. Finally, we compute a total of 288 sentence-level features based on word posterior probabilities.

Naïve Bayes' Classifier Features *(B, 72 features, [Sanchis et al., 2007])*

In addition to the raw values of the features described in the previous section, we also use them to build a smoothed naïve Bayes' classifier that estimates the probability of each word to be a correct translation [Sanchis et al., 2007].

The class variable is denoted by c ($c = 0$ for incorrect and $c = 1$ for correct). Given a target word, e, and a vector $\mathbf{z} = (z_1, \ldots, z_d, \ldots, z_D)$ of features, the class posterior can be calculated via the Bayes' rule as:

$$\mathrm{P}(c \mid \mathbf{z}, \mathrm{e}) = \frac{\mathrm{P}(c \mid \mathrm{e}) \cdot \mathrm{P}(\mathbf{z} \mid c, \mathrm{e})}{\sum_{c'} \mathrm{P}(c' \mid \mathrm{e}) \cdot \mathrm{P}(\mathbf{z} \mid c', \mathrm{e})} \tag{3.5.8}$$

For simplicity, the model includes the naïve Bayes' assumption that the features are mutually independent given a class-word pair:

$$\mathrm{P}(\mathbf{z} \mid c, \mathrm{e}) = \prod_{d=1}^{D} \mathrm{P}(z_d \mid c, \mathrm{e}) \tag{3.5.9}$$

Thus, the basic problem is to estimate $\mathrm{P}(c \mid \mathrm{e})$ for each target word, and $\mathrm{P}(\mathbf{z} \mid c, \mathrm{e})$ for each class-word pair. Given N training samples $\{(\mathbf{z}_n, c_n, \mathrm{e}_n)\}_{n=1}^{N}$ these probability distributions can be estimated via relative frequencies:

$$\mathrm{P}(c \mid \mathrm{e}) = \frac{\#(c, \mathrm{e})}{\#(\mathrm{e})} \qquad \mathrm{P}(z_d \mid c, \mathrm{e}) = \frac{\#(z_d, c, \mathrm{e})}{\#(c, \mathrm{e})} \tag{3.5.10}$$

where $\#(\cdot)$ are event counts of different events: single words (e), (c, e) pairs, or (z_d, c, e) triplets.

In the experiments, we used the N-best posterior probability features described above as input for the naïve Bayes' classifier. However, these features have continuous rather than discrete domains. Thus, to use Equations (3.5.10) a discretization of the features is required. This is done by dividing the feature domain into a fixed number of evenly-spaced bins of fixed size. The bin size is selected so that it maximizes classification accuracy in a separated development set.

Additionally, to avoid learning problems with rare events, the model is smoothed using absolute discounting. The idea is to discount a small constant $b \in (0, 1)$ to every positive count, and then to distribute the gained probability mass among the null counts. A detailed explanation of absolute discounting can be found in [Ney et al., 1997].

Once the parameters of the model are estimated, a target word e is classified as correct if the confidence estimation $\mathrm{P}(c = 1 \mid \mathbf{z}, \mathrm{e})$ is greater than

a certain threshold τ. The value of τ is also selected so that it maximizes classification accuracy in a separated development set.

In order to train and evaluate this model, a corpora is needed where each word is tagged as correct or incorrect. This can be done manually by human experts but it is a very time-consuming task. Hence, an automatic tagging of the words using reference translations is carried out instead. We consider three different tagging methods [Sanchis et al., 2007]:

- Each word is tagged as correct if it is Levenshtein aligned to itself in the reference.
- Each word is searched in the whole reference and, if found, it is drawn *without* replacement and tagged as correct.
- Each word is searched in the whole reference and, if found, it is drawn *with* replacement and tagged as correct.

Thus, for each word we can compute three different features $\mathrm{P}(c = 1 \mid \mathbf{z}, \mathrm{e})$ depending on the tagging method selected. Moreover, we also extend these features to subsequences similarly as done in the previous section. Obviously, these automatic tagging methods are not perfect but they are faster, easier, and cheaper than performing a manual tagging.

Finally, we represent each subsequence-feature by the five sentence level indicators described at the beginning of the Section 3.5.3, accounting for 60 sentence level features (four sizes $\times$ three tagging criteria $\times$ five indicators). We compute an additional feature denoting the number of sequences classified as correct by the model, 12 features (four sizes $\times$ three tagging criteria), for a total of 72 sentence level features.

3.6 Experiments

3.6.1 Evaluation Criteria

Since we view DR as a way to build robust prediction models, we evaluated each DR method by the prediction accuracy of the regression models trained on the corresponding reduced feature sets. Given a test corpus with N samples, the performance of a regression model is usually measured by the average error of the predicted quality scores $\hat{\mathbf{y}} = \{\hat{y}_1, \ldots, \hat{y}_n, \ldots, \hat{y}_N\}$ with respect to the actual reference scores $\mathbf{y} = \{y_1, \ldots, y_n, \ldots, y_N\}$. Specifically, we compute the

root mean squared error (RMSE) between them as in [Specia et al., 2009b,a]:

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2} \tag{3.6.1}$$

RMSE quantifies the average deviation of the estimation with respect to the expected score. I.e. the lower the value, the better the performance of the learning model.

In the shared QE task featured at the 2012 workshop on SMT [Callison-Burch et al., 2012], the main evaluation measure was the mean absolute error (MAE) of the predictions while RMSE was used as a secondary evaluation measure. Certainly, both MAE or RMSE could be used. However, we chose to use RMSE due to some interesting properties it possesses. For example, RMSE has a direct relation to the coefficient of determination [Steel and Torrie, 1960] (R^2) that is widely used in statistics to measure how well a regression model fits a set of data. Additionally, RMSE values have a more intuitive interpretation than R^2 values for this task, namely the actual magnitude of the average prediction error of the system.

3.6.2 Experiments to Determine the Best Configuration of the Proposed Training Methodology

We extracted the 480 features described in Section 3.5 for each of the automatic translations in the evaluation data of the QE task. As a result, we obtained a training set and a test set of 480-dimensional real vectors with 1832 and 422 samples respectively. All features were standardized by subtracting the feature mean from the raw values, and dividing the difference by the standard deviation.

Then, we carried out an exhaustive experimentation to test the different DR methods described in Section 3.3, and to study how their use affect the prediction performance of the different learning models presented in Section 3.4. We tested all 18 combinations of a DR method and a learning model in a series of two-step experiments as depicted in Figure 3.1. Since we did not know the optimum size r of the reduced feature set (see Section 3.3.2), each experiment involved several trains of the model with reduced feature sets of different sizes. For each size, we performed a cross-validation training with a ten-fold random partition of the 1832 training samples to learn the meta-parameters of the models, e.g. the γ parameter of ridge regression. We used eight folds for training, one separated fold for development, and report results on another separated test fold. The optimal values of the meta-parameters are then used

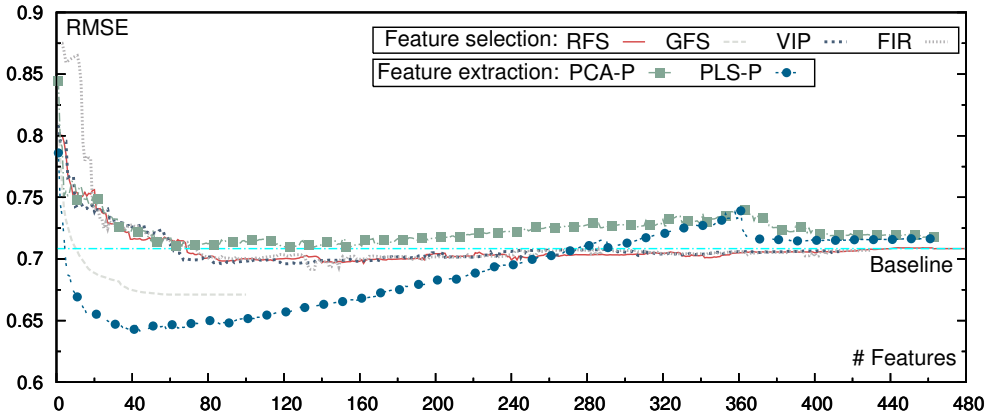


Figure 3.5: SVMs cross-validation training results for different DR methods as a function of the size of the reduced feature set. In comparison, the baseline SVM trained on the 480 original features obtained 0.71 RMSE. Best PLS-P results were statistically better than the rest.

to train a model with the complete training set. Finally, we used this optimized model to predict quality scores for the held-out test set.

Cross-Validation Training Results

We now present the results for the cross-validation training experiments. The final conclusions were similar for all learning models. Thus, to keep the presentation clear, we focus the discussion on the results using SVMs as learning model. Figure 3.5 displays the SVMs cross-validation RMSE for the different DR methods presented in Section 3.3 as a function of the size of the reduced feature set.

Let us comment first the results for the four feature selection methods under study. Rank of feature selection (RFS), variance importance in projection (VIP), and feature importance in regression (FIR) obtained virtually the same results, slightly outperforming the baseline SVM model trained with the whole 480-dimensional feature set (0.71 RMSE). Their performance improved as more features were selected, and they required to select above 100 features to reach their top performance. Then, as more features were selected their results slowly converged to the performance of the baseline model. Since these methods do not take into account the correlations that may exist between the features, their reduced feature sets were highly-redundant; which explains the large number of features they needed to stabilize. In contrast, greedy for-

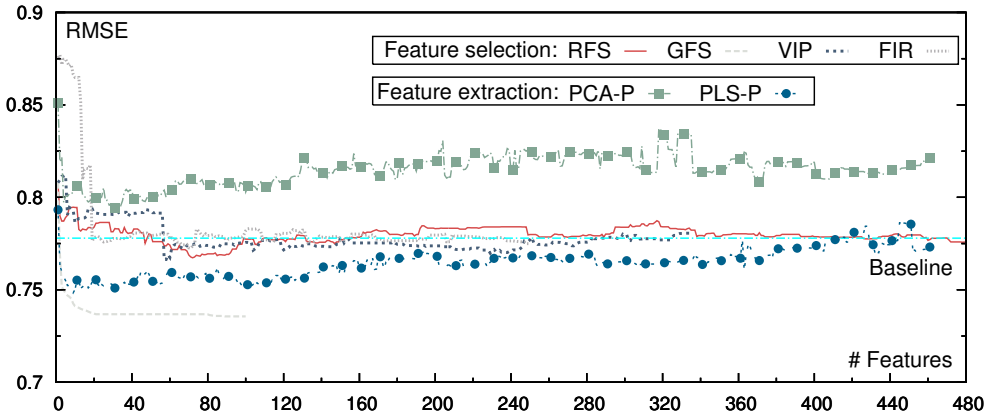


Figure 3.6: M5P cross-validation training results for different DR methods as a function of the size of the reduced feature set. The baseline M5P model trained on the 480 original features obtained 0.78 RMSE.

ward selection (GFS) was able to clearly outperform the baseline with few features. However, its higher computational complexity complicates its practical deployment. This computation complexity is the reason why we carried out experiments only up to 100 features where its performance seemed to stabilize. Nevertheless, with only 13 features it was able to equal the performance of the baseline model trained on the original 480 features.

Regarding the two feature extraction methods, they exhibited important differences in performance. PCA projection (PCA-P) obtained worse results than the four feature selection methods, moreover it did not even improve the results of the baseline model. PCA-P reached its top performance when ~ 120 principal components were generated, and it slightly deteriorated as the number of features increased. In contrast, PLSR projection (PLS-P) obtained much better results consistently outperforming PCA-P and all feature selection methods. Moreover, with only five latent variables PLS-P was able to outperform the baseline SVM model trained with 480 features, and it only required 44 features to reach its top performance. Additionally, the performance difference observed between the best result of PLS-P and the rest DR methods was significant with a probability of improvement of 95% according to a pair-wise bootstrap analysis [Bisani and Ney, 2004]. These results indicate that PLS-P generates more "information-dense" features that constitute a better summary of the original high-dimensional feature set.

Although results in Figure 3.5 are representative for all learning models, we observed important differences in the stability of the learning curves of

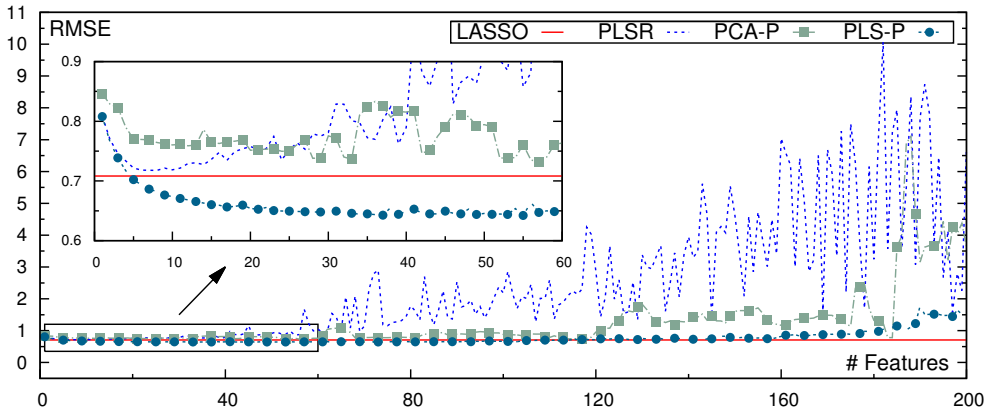


Figure 3.7: Cross-validation training results for linear ridge regression using PCA-P and PLS-P DR methods. We also display baseline results for LASSO regression, and PLSR (Equation (3.3.4)). As for SVMs in Figure 3.5, PLS-P outperforms any other tested approaches. Additionally, note that the use of the proposed two-step training procedure, see Figure 3.1, allows to smooth the rough learning curves obtained by conventional PLSR, compare PLSR and PLS-P learning curves.

the different models. Figure 3.6 displays cross-validation training results by M5P regression trees for the different DR methods under study. Again, due to the higher complexity of GFS, we only carried out experiments up to 100 selected features. We observed that the learning curves were slightly noisier, and differences in performance were scarcer than the ones obtained by SVMs. Additionally, in this case GFS slightly outperformed PLS-P. These facts seemed to indicate that feature extraction methods (PCA-P and PLS-P) are not particularly adequate to be used together with M5P regression trees. Since the features generated by feature extraction methods are the combination of several of the original features, we hypothesize that they are also more difficult to be partitioned into regions to create the tree structure of the model.

Finally, Figure 3.7 displays training cross-validation results for linear ridge regression using PCA-P and PLS-P as DR methods. We present results only for these two DR methods for simplicity. Since the baseline ridge model (trained with the original 480 features) obtained a dreadful RMSE of 16.73, we present results for two alternative linear regression baselines: a LASSO regression model (see Section 3.4.1) also trained with all the original 480 features, and for the predictions directly generated by the PLSR model according to Equation (3.3.4). In contrast to the results for SVMs and M5P, we obtained much noisier learning curves with large performance variations, particularly as we

	Ridge regression		Support vector machines		Regression trees	
	RMSE	#features	RMSE	#features	RMSE	#features
Original features	0.79	480	0.82	480	0.87	480
RFS	0.83	69	0.84	162	0.91	72
GFS	0.80	61	0.80	100	**0.86**	89
VIP	0.83	67	0.83	126	0.88	57
FIR	0.83	82	0.82	136	**0.86**	71
PCA-P	0.83	57	0.81	122	0.90	31
PLS-P	**0.78**	55	**0.78**	44	0.88	9

Table 3.2: Prediction results (RMSE) on the test set for the different DR methods and learning models under study. #features denotes the number of features of the reduced test sets. Best results for each learning model are displayed in bold. As a comparison, the result for a linear LASSO regression model was 0.82 RMSE. The top performing system [Soricut et al., 2012] submitted to the shared QE task of the 2012 workshop on SMT scored 0.75 RMSE.

increased the number of features. However, the proposed two-step training procedure (see Figure 3.1) partially addresses this problem. This is exemplified in the comparison between PLSR and PLS-P. Both methods use a linear model to predict the quality scores from the projected data, however PLS-P obtains a much smoother learning curve than PLSR. Finally, we could extract the same conclusion as for SVMs: among all the tested DR methods, PLS-P is the best performing one allowing us to improve the performance of even sophisticated regularized models such as SVMs or linear LASSO regression.

These results show that the proposed two-step training is an efficient procedure to deal with noisy and correlated input features, and that it can outperform models such as LASSO regression and PLSR that also integrate DR in their formulation.

Blind Test Results

Next, for each combination of a DR method and a learning model, we built a new model using the full training set and the best configuration (size of the reduced feature set, and values of the meta-parameters of the learning model) observed in the corresponding cross-validation experiment. Then, we reduced the held-out test set using the same DR method as in training, and tested the

performance of the optimized model for the reduced test set. Table 3.2 displays these results. In contrast to the previous cross-validation experiments, results on the test set were quite different for the three learning models. While for SVMs, the use of DR improved the performance of the baseline model trained on the 480 original features, no improvement was obtained at all for linear ridge regression, or for regression trees. This was quite a surprising result. Given the large improvements over the baseline obtained in the cross-validation experiments, we expected to obtain similar improvements over baseline in test.

To better understand these results, we carried out a multivariate Hotelling's two-sample T-squared test [Hotelling, 1931; Anderson, 1958] to study the possible differences that may exist between the training and test partitions of the feature set. The objective of such tests is to determine whether two samples, in our case the training and test sets, have been sampled from the same population or not. The result of the test indicated that there were a statistically significant difference between the two feature sets ($p < 0.01$), and thus they seemed to come from different populations. Since the training and test translations come from a similar news domain [Callison-Burch et al., 2012], we hypothesize that the difference between the feature sets was due to the specific chosen features. In fact, results of individual Student's two-samples t-tests [Gosset, 1908] for each feature showed that 260 of the 480 extracted features were significantly different ($p < 0.01$) between training and test. For example, the number of words with zero posterior probability is significantly different between the samples in training ($\mu = 1.7, \sigma = 1.4$) and test ($\mu = 0.9, \sigma = 0.8$).

The results of these statistical tests seemed to confirm our intuition that the training set does not adequately represents the test set. However, the baseline systems seemed to be less affected by these fact than the systems build following the proposed training methodology. This can be answered by reviewing the proposed training methodology (see Figure 3.1). As a first step, the proposed training methodology projects the original data into a new space. Of course in training-time the system only has access to the training partition of the data, hence this projection is computed based only in the information contained in the training partition. Therefore, if the training partition is not representative of the test partition, as it seems to be the case, the reduced feature sets will be projected in a "direction" that cannot be adequate to improve prediction accuracy in the test set. In other words, features that are irrelevant or redundant given the training partition, may be crucial to accurately predict the test data. In other words, since the proposed method relies in training data to strip out the redundancy and noise present in the

original data, it may eliminate information necessary to accurately predict the quality scores of the test partition. Additionally, test RMSE results show that given the scarce data available cross-validation training is not sufficient for the proposed approach to address this problem. It is worthy of notice that this drawback is not specific of the chosen DR method, but it is a characteristic common to all DR techniques.

The fact that SVMs actually improved baseline test results when DR methods were used can be explained by the fact that SVMs are more complex models than ridge regression and regression trees. SVMs performance is more heavily penalized due to the lack of data. Thus, we hypothesize that the use of reduced feature sets, even if they are inadequate, allows to improve SVMs performance[h]. Despite these problems, Table 3.2 shows that PLS-P was the top-performing DR method for linear regression and SVMs. However, for regression trees, all methods obtained similar results. This fact confirms the results obtained in the cross-validation experiments (see Figure 3.6), that regression trees were not adequate to fully exploit the more "information-dense" features generated by PLS-P.

Nevertheless, even in this pessimistic setting PLS-P generated reduced feature sets that performed similarly as the original 480 features. Moreover, we were able to obtain similar results (0.78 RMSE versus 0.75 RMSE) as the best-performing system [Soricut et al., 2012] submitted to the shared QE task of the 2012 workshop on SMT [Callison-Burch et al., 2012]. We consider that, given the cross-validation results in Section 3.6.2, larger performance improvements could be expected whenever an adequate set of features, and/or a large enough training set are provided.

Lastly, since the time required to train the model and to perform the prediction is directly related to the number of features, an additional advantage of the proposed methodology is that it can improve the practical deployment of QE technology by reducing training/test time. For example, training an SVM (including meta-parameter optimization) using the original 480 features typically required $\sim$30 hours in our test machine, while the training time using the optimal 44 latent variables extracted by PLS-P was below three hours.

Feature Analysis

To finalize this series of experiments, we perform an additional analysis on which are the features that contribute more to create the reduced feature sets. For feature selection methods, we simply looked for the most frequently se-

[h] Few features imply few parameters to be estimated with the same amount of data.

lected features. For PCA-P and PLS-P, that combine the original features into new features (the principal components and the latent variables respectively) by a matrix transformation (**P** in Equations (3.3.2) and (3.3.6)), we computed the contribution of each feature by summing up the absolute value of the scores in the corresponding column of **P**. We then can highlight the following features:

- Source and translation lengths and language model probabilities.
- Vocabulary of the 1000-best translations divided by their average length.
- Number of source phrases of size one used in decoding.
- Number of source phrases used in decoding.
- Frequencies of source subsequences (sizes one to four). †
- Posterior probabilities of translation subsequences (sizes one and two).†
- Probability of the translation subsequences (sizes one and two) by a naïve Bayes' classifier.†

Specifically, for the subsequence-based features (marked with †) the most important sentence-level indicators were the average value of the scores of the subsequences in the translation, and the number of them belonging to the first and fourth quartile.

We also observed slight differences in the importance of each feature according to the different DR methods. For example, the simple RFS, tended to add lots of similar features which independently are quite informative but together are highly redundant. In contrast, the most computationally complex method, GFS, selected only one or two features that represent all features of the same type.

3.6.3 Exhaustive Experiments with Several Feature Sets

Once we have determined that the use of PLS-P with SVMs is the best-performing setup of the proposed two-step QE training algorithm, we carried out a new series of experiments intended to exhaustively evaluate this particular setup. To do that, we computed quality scores with the feature sets used by the different QE systems submitted to the shared QE task [Callison-Burch et al., 2012][i]. These feature sets allow us to test our approach under a wide variety of conditions in terms of number of features, previous application of feature selection techniques, redundancy, and noise (features irrelevant for the

[i]The feature sets used by the different systems submitted to the task are publicly available in: https://github.com/lspecia/QualityEstimation.

prediction). Since our focus in on the training process, we consider the feature sets as independent corpora provided by an external agent. For this reason, we only provide a brief description of the eight sets used in the experimentation. An exhaustive description of each set can be found in the corresponding citation. Many of the sets include the 17 baseline features provided by the organizers of the shared QE task [Callison-Burch et al., 2012].

DCU-SYMC: [Rubino et al., 2012] The set contains 308 features including features computed using latent Dirichlet allocation, source grammatical features extracted using the TreeTagger part-of-speech tagger, an English grammar, the XLE parser, and the Brown re-ranking parser; and target features based on part-of-speech tag counts extracted using a Spanish TreeTagger model.

LORIA: [Langlois et al., 2012] The set contains the baseline features, and a number of features proposed by the authors in previous works, amounting for a total of 66 features.

SDLLW: [Soricut et al., 2012] The set contains 15 features selected from an original set of 45 features: the 17 baseline features, 8 system-dependent features from the decoder information, and 20 features developed internally by the authors. An exhaustive feature selection process that directly optimize the metrics used in the task was followed to select the final set of features.

TCD: [Moreau and Vogel, 2012] The set contains 43 features including the baseline features and features based on distances. The latter features work in the following way: given a sentence to evaluate, it is compared against some reference sentence using similarity measures, and the obtained score is then used as a feature. The training data and the Google n-grams dataset were used as references.

UEDIN: [Buck, 2012] The set contains 56 features that include the baseline features and features based on named entities, binary indicators of punctuation, word in upper case and numbers, target lexicon probabilities, word-alignments between source and target, n-gram counts, and distance to the closest sentence in the training corpus.

UPV: [González-Rubio et al., 2012] The set contains 497 features, including the baseline features and the features described in section 3.5.

UU: [Hardmeier et al., 2012] The set contains 82 features computed from syntactic, constituency, and dependency trees.

Name	#features	feature selection?	collinear features	constant features
DCU-SYMC	308	no	34.6%	0.7%
LORIA	49	yes	12.2%	0.0%
SDLLW	15	yes	0.0%	0.0%
TCD	43	no	18.6%	0.0%
UEDIN	56	no	5.5%	1.8%
UPV	497	no	54.3%	6.8%
UU	82	no	7.5%	2.5%
WLV-SHEF	147	no	21.0%	2.7%

Table 3.3: Main properties of the different sets of features. We estimate the degree of collinearity of each feature by its condition number as described in [Cheney and Kincaid, 2012].

WLV-SHEF: [Felice and Specia, 2012] The set contains 147 linguistically-informed features including features computed from part-of-speech information, phrase constituency, subject-verb agreement, and target lexicon analysis.

Table 3.3 displays, for each set, the number of features, whether or not the features have been obtained after a feature selection process, the percentage of features in the training partition that are redundant, i.e. that are collinear with the rest of features, and the percentage of features in the training partition that are constant, and hence, irrelevant to perform the prediction. We estimate the degree of collinearity of each feature by its *condition number* considering a value above 100 to denote collinearity [Cheney and Kincaid, 2012].

Each of the above described feature sets defines equivalent training and test partitions with 1832 and 422 samples respectively. All features were standardized by subtracting the feature mean from the raw values, and dividing the difference by the corresponding standard deviation.

For each feature set, a QE system was built with the optimum setup (PLS-P as DR method followed by SVMs for prediction, see Section 3.6.2) of the two-step methodology depicted in Figure 3.1. All features were standardized by subtracting the feature mean from the raw values, and dividing the difference by the corresponding standard deviation.

Free parameters, namely the number of latent variables (r) and the SVM meta-parameters (γ, ϵ, and C) were optimized by ten-fold cross-validation using the training partitions (1832 samples). Each cross-validation experiment considered eight folds for training, one held-out fold for development and the

Feature set	Baseline		PCA-P		PLS-P	
	RMSE	#feat.	RMSE	#features	RMSE	#features
DCU-SYMC	0.71±0.02	308	0.70±0.02	82 (26.6%)	**0.62±0.02***	**28 (9.1%)**
LORIA	**0.72±0.03**	49	0.75±0.01	43 (87.7%)	**0.72±0.02**	**10 (20.4%)**
SDLLW	**0.67±0.02**	15	**0.67±0.02**	15 (100.0%)	**0.67±0.02**	**10 (66.7%)**
TCD	0.76±0.01	43	0.74±0.02	24 (55.8%)	**0.72±0.02**	**15 (38.9%)**
UEDIN	0.72±0.03	56	0.71±0.02	43 (76.8%)	**0.69±0.02**	**8 (14.3%)**
UPV	0.74±0.02	497	0.69±0.02	99 (19.9%)	**0.62±0.02***	**58 (11.7%)**
UU	0.72±0.02	82	0.68±0.02	74 (90.2%)	**0.67±0.02**	**29 (35.4%)**
WLV-SHEF	0.71±0.02	147	0.71±0.02	91 (61.9%)	**0.65±0.02***	**25 (17.0%)**

Table 3.4: RMSE and optimum number of latent variables obtained by cross-validation for the different feature sets. Results are the average over the ten held-out test folds. In parenthesis, we display the number of latent variables as a percentage of the original features. Baseline denotes the results of a system trained with the whole feature set. Best mean RMSE result and lowest number of features are displayed boldface. Asterisks indicate a statistically better result than the *both* the other two systems at 95% confidence.

other held-out fold for test. We used the training folds to estimate a PLS model which was then used to extract the r latent variables of the training, development and test folds. Then, we used the latent variables of the training folds to estimate an SVM prediction model, the reduced development fold to optimize the SVM meta-parameters (γ, ϵ, and C), and the reduced test fold to test the optimized SVM model. The result of each complete cross-validation experiment was the average of the performance on the ten held-out test folds. The number of latent variables was selected to optimize this average prediction accuracy.

Once the number of latent variables was fixed, we trained a new prediction model with the whole training partition optimizing the SVM meta-parameters by cross-validation. Finally, we used this optimized SVM model to predict the quality scores of the corresponding test partitions (422 samples).

Next, we present the results of the experimentation performed to evaluate the proposed QE approach. First, we conducted a series of experiments in a classical QE scenario where we predicted quality scores from the knowledge-based feature sets described in the previous section. Then, we took advantage of the scalability of the proposed methodology to predict quality scores using an extremelly large and noisy feature set that includes all the features in the sets previously described.

Results for Predictions Computed from Feature-Engineered Sets

Table 3.4 shows the results (RMSE and optimal size of the reduced feature set) obtained by PLS-P in cross-validation training experiments for the different feature sets. As a comparison, we present baseline results for SVMs trained with all the features in each set, and results using the widespread PCA-P method instead of PLS-P to reduce the feature sets.

We can observe that PLS-P consistently obtained equal or better prediction accuracy (RMSE) than the baseline systems. Additionally, the optimum number of features employed to build the final SVMs using PLS-P was much lower than the number of original features. The size of the reduced sets varied between two thirds and one tenth of the original features. These reductions are roughly related with the percentage of collinear and constant features in Table 3.3. In comparison to PCA-P, PLS-P was able to obtain better prediction accuracy with less features. Usually, the number of latent variables was less than half the number of principal components.

These result indicate that the proposed QE approach was indeed able to strip out the noise present in the original feature sets. Additionally, the proposed DR technique based on PLS showed a better performance (both in prediction accuracy and reduction ratio) that the commonly-used PCA. As a result, even for highly-engineered feature sets such as SDLLW [Soricut et al., 2012] that contain almost no collinear or redundant features, our approach was able to obtain a more compact feature set (10 latent variables) that still retained the prediction potential of the whole original set (15 features).

Next, to better understand the influence of the number of reduced features R in the results, we display in Figure 3.8 a detailed graph of the prediction accuracy curves obtained for two prototypical feature sets: the highly noisy and collinear UPV set, and the low redundant SDLLW set.

The prediction accuracy of PLS-P for the UPV feature set (top panel in Figure 3.8) rapidly improved as more latent variables were considered. With only five latent variables, prediction accuracy already outperformed the baseline (497 features), and it reached its top performance for 58 latent variables. As we considered more latent variables, for simplicity the graph only shows up to 100 features, prediction error steadily increased which was indicative of over-training. Thus, we chose 58 as the optimum number of variables for the UPV set. The quite large RMSE improvement in comparison to the baseline can be explained by the ability of our approach to strip out the great amount of noise present in the original UPV set, see Table 3.3. Regarding PCA-P, it was consistently outperformed by PLS-P and only slightly improved the RMSE score of the baseline system.

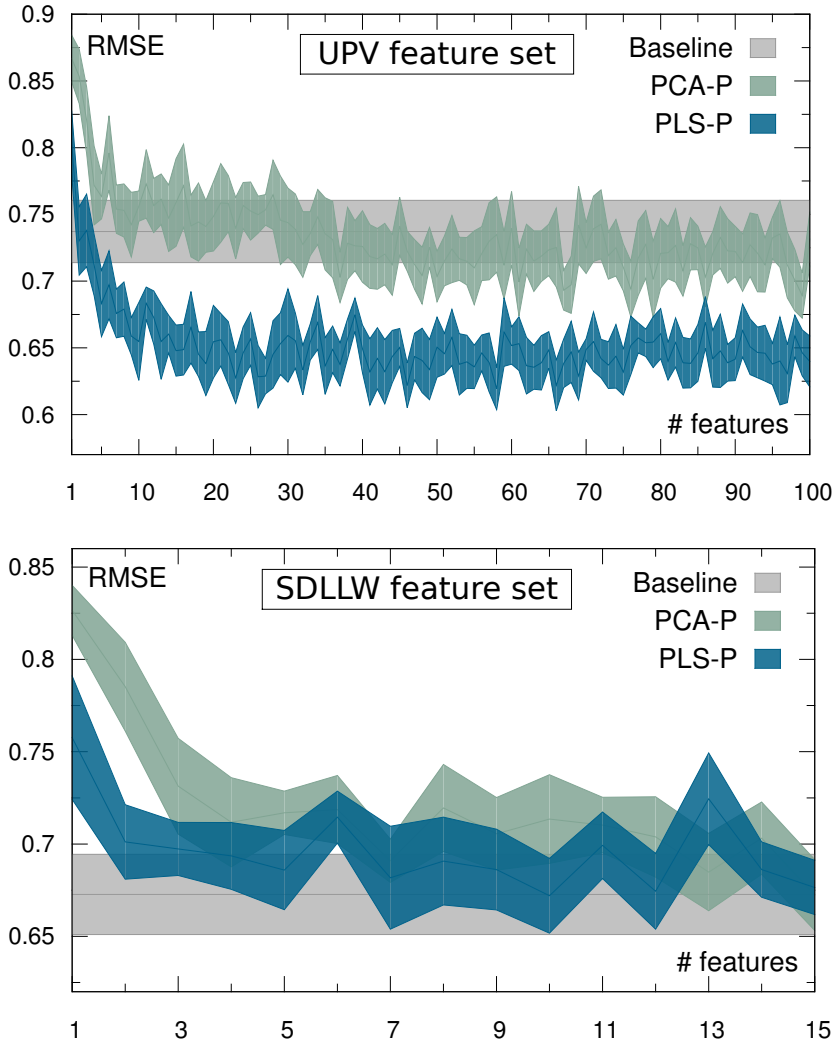


Figure 3.8: Cross-validation prediction accuracy curves (RMSE and 95% confidence interval) for two representative feature sets: the highly-redundant UPV set (above), and the concise SDLLW set (below). Baseline denotes the RMSE of systems trained with the whole original feature sets: 497 features for UPV set, and 15 features for SDLLW set.

For the low redundant SDLLW feature set (bottom panel in Figure 3.8), PLS-P showed approximately the same behavior: prediction accuracy rapidly improved up to a point from where the performance steadily deteriorated. In this case, ten was the optimal number of latent variables. In contrast to

Feature set	Baseline	PCA-P	PLS-P
DCU-SYMC	**0.87±0.07***	1.01±0.07	0.96±0.08
LORIA	**0.84±0.06**	0.87±0.06	0.85±0.06
SDLLW	**0.76±0.05**	0.77±0.05	**0.76±0.05**
TCD	**0.82±0.06**	1.00±0.05	0.83±0.06
UEDIN	0.86±0.06	**0.85±0.05**	0.86±0.05
UPV	0.82±0.06	0.83±0.05	**0.78±0.05***
UU	**0.81±0.05**	**0.81±0.05**	0.82±0.06
WLV-SHEF	0.84±0.05	0.84±0.05	**0.82±0.05***

Table 3.5: RMSE and 95% confidence intervals of the predictions for the test partitions. For all feature sets, confidence intervals overlap. However, the observed differences were significant for some feature sets according to paired bootstrap re-sampling. Best average results are displayed boldface. Asterisks denote a statistically significant difference in performance with respect to *both* the other two methods. Significance was measured by paired re-sampling with 95% confidence.

the UPV set, our approach could not improve Baseline performance which is reasonable since SDLLW is a very clean set with no redundant or irrelevant features (see Table 3.3) that could hinder the learning process of the baseline model. Nevertheless, PLS-P was able to obtain the same prediction accuracy as Baseline with only two thirds the number of the original features.

In the following experiment, we built QE systems with the whole training partitions and the optimal size of the reduced feature set T estimated in the cross-validation experiments. The SVM meta-parameters (γ, ϵ, and C) were optimized by standard cross-validation and the optimized models were used to predict the quality scores of the held-out test partitions. Note that due to variations in the learning procedures, Baseline results may differ from those reported in the WMT12 QE task [Callison-Burch et al., 2012] .

Table 3.5 displays, for each feature set, the RMSE obtained by PLS-P in the test partition. We also show baseline results for SVMs built with all the features in each set, and for systems that used PCA instead of PLS to obtain the reduced feature sets (PCA-P). Confidence intervals for the RMSE results of Baseline, PCA-P and PLS-P always overlapped but the observed differences were still statistically significant for a number of sets: for DCU-SYMC, Baseline obtained a statistically better result than PCA-P and PLS-P; for LORIA and TCD, no statistically significant difference was observed between PLS-P and Baseline but both systems obtained a statistically better result than PCA-P; for UPV and WLV-SHEF, PLS-P statistically outperformed the

DCU-SYMC	45.1%	UEDIN	48.1%
LORIA	24.5%	UPV	67.4%
SDLLW	73.3%	UU	38.8%
TCD	30.2%	WLV-SHEF	28.6%

Table 3.6: Percentage of the features in each feature set that exhibit significantly different values between the training and test partitions. Significance computed by Student's two-sample t-test ($p<0.01$).

other two methods; and for SDLLW, UEDIN and UU, no significant differences were found between the three systems.

As for the previous experiments in Section 3.6.2, these test results were quite surprising. Given the significant RMSE differences observed in cross-validation, see Table 3.4, we expected to obtain similar improvements over Baseline in test. We followed a careful cross-validation training process (see Section 3.6.3) where each experiment was evaluated in a held-out test fold used not to reduce the dimensionality nor to estimate the prediction model. Therefore, we again hypothesize that the explanation for the results in Table 3.5 was that the training partitions were not representative of the test partitions. We again carried out multivariate Hotelling's two-sample T^2 tests [Hotelling, 1931; Anderson, 1958] to evaluate this hypothesis and found that indeed training and test partitions were statistically different for all feature sets ($p<0.01$). Then, we performed a series of Student's two-sample t-tests to study the possible difference between the values of the features in each set. The results of these tests indicated that most of the features did exhibit statistically different values ($p<0.01$) between training and test. Table 3.6 displays the actual percentage of the features in each set that have significantly different values between the training and test partitions.

This mismatch can be partially explained by the fact that the training and test partitions were extracted from news texts of different years [Callison-Burch et al., 2012], but we still consider that the main problem is the size (only 1832 samples) of training partitions that did not adequately represent test partitions. Better RMSE scores, see Table 3.4, can be expected if adequate training partitions are provided. However, both PLS-P and the baseline systems had to deal with this mismatch, so, why PLS-P seemed to be more heavily penalized than the baseline system?

This question can be answered by reviewing the proposed methodology. In training time, the DR method only has access to the training data to project

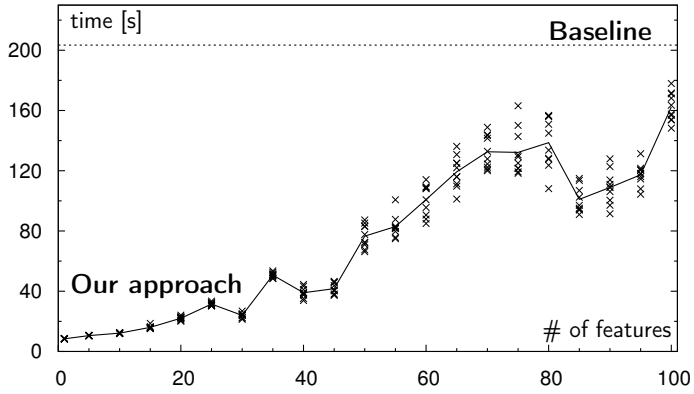


Figure 3.9: Operating time (training plus prediction) of the SVM regression model as a function of the number of features used to built the QE system. Baseline system was trained with the 147 original features of the WLV-SHEF set.

the features into a new space. Therefore, if the training partition is not representative of the test partition, the reduced feature sets will be projected in a "direction" that cannot be adequate to improve prediction accuracy in the test set. I.e., crucial information to predict the quality scores of the test partition may be stripped out. Again, this drawback is not specific of the chosen DR method, but it is a characteristic common to all DR techniques. This fact is exemplified in Table 3.5 by the test results obtained using PCA-P instead of PLS-P.

Nevertheless, even in this pessimistic setting, the PLS-P was able to obtain comparable results to the baseline system, as shown by the overlapping RMSE confidence intervals in Table 3.5, and, given the encouraging cross-validation results in Table 3.4, larger performance improvements could be expected in test whenever an adequate training partition is provided.

Additionally, the main advantage of the proposed approach, is that fewer features imply lower operating times for the QE system. Figure 3.9 displays the time required to build a SVM model (including meta-parameter optimization) and obtain the predictions for the test set as a function of the number of features used to train the model. Specifically, we built QE systems with an increasing number of latent variables from the WLV-SHEF feature set. Each point in the figure is the average time of ten experiments. Results show how operating times increased with the number of latent variables. For instance, the operating time of the baseline model trained with the original 147 features

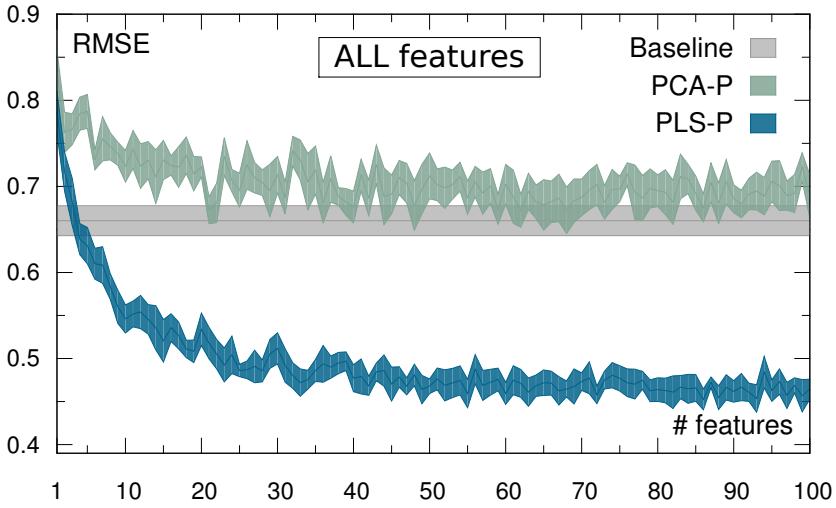


Figure 3.10: Cross-validation prediction accuracy curve for the high-dimensional (1197 features) ALL feature set.

was $\sim$200 seconds, while the operating time of the system built with the 25 latent variables selected by PLS was only $\sim$35 seconds which represents almost one order of magnitude less operating time. Therefore, our approach is well-suited to be implemented in scenarios with strict temporal restrictions, such as the interactive MT framework in Chapter 4.

Exploiting the Scalability of our Methodology

Previous experiments have shown that PLS-P is quite efficient in summarizing the relevant information contained in the original noisy features. Thus, we now present results for an scenario where all the features used in the previous experiments are joined together to create an high-dimensional feature set from which to predict quality scores. This aggregated set, denoted by ALL, contains 1197 features for each translation; approximately 55% of them being collinear with the rest. ALL can be seen as a simulation of a QE scenario with no feature engineering, i.e., an scenario where every attribute available is measured in the hope that the relevant variables can be automatically isolated.

Figure 3.10 shows cross-validation prediction accuracy (RMSE and 95% confidence interval) of PLS-P as a function of the number of reduced features. Again, we also display results for a baseline SVM model built using all the features, and for a system built using PCA-P instead of PLS-P to reduce the

dimensionality. Results show that PLS-P was able to largely improve Baseline results using only a small number of latent variables. We obtained a score of 0.45 ± 0.01 RMSE with only 86 features. This result represents approximately a 30% reduction over the RMSE of the baseline system that was trained with 1197 features. Regarding PCA-P, it barely reached the performance of the baseline system. These results indicate that PLS-P was able to adequately exploit the information contained in the ALL set to improve prediction accuracy. In contrast, both the baseline systems and PCA-P were unable to manage the large number of noisy and collinear features. Additionally, the operating time of the baseline systems was $\sim$23 minutes while it reduced to $\sim$2 minutes when we used the optimal 86 reduced features extracted by PLS-P.

However, for the same reasons we have described above, the results of PLS-P for the test set were again quite disappointing: 1.4 ± 0.1 RMSE versus 0.78 ± 0.06 RMSE of the baseline system and 0.81 ± 0.07 of PCA-P. Note that contrary to the previous experiments, here our approach obtained a clearly worse result than baseline and PCA-P. We hypothesize that this was due to the larger number of original features available. As more features relevant for the prediction are available, PLS-P is able to generate more "specialized" latent variables. Given that the training data does not adequately represents the test data (see discussion above), this better projection (as shown in Figure 3.10) actually hinders prediction accuracy in the test set.

3.7 Summary

We have presented a two-step training methodology developed to address the learning issues inherent to the use of noisy and collinear features such as the ones employed to estimate the quality of automatic translations. The cornerstone of our approach is the DR method that is in charge of stripping out the redundancy present in the features and generate a reduced feature set suitable to train robust QE systems. Additionally, we have proposed two novel DR methods based on PLSR and compared them against several DR methods previously used in the QE literature. The DR methods under consideration can be classified by their theoretical background: statistical multivariate analysis or heuristic methods, or by how they perform the reduction: feature selection or feature extraction methods. Moreover, we have studied how DR affect the prediction performance of different learning models.

We have evaluated each DR method by the prediction performance of the learning models trained on the corresponding reduced feature set. This quality measure has the advantage of automatic evaluation, and, using identical

pipelines to train the models, it allows us to accurately compare the different DR methods. The key results of the experiments are as follows:

- Feature extraction methods can outperform feature selection methods.
- Statistical methods based on multivariate analysis can outperform heuristic methods.
- To obtain a good prediction performance, DR methods have to take into account the scores to be predicted.
- The performance-wise ranking of the DR methods is to a great extent independent of the chosen learning model.
- However, for simpler models such as linear regression the use of some DR methods may result in erratic learning curves.

One of the proposed DR methods, PLS-P, has all the desirable properties according to these conclusions: it is a feature extraction method based on multivariate analysis that takes into account the values to be predicted to perform the reduction.

Empirical results with multiple feature sets have shown that PLS-P was able to obtain large feature reduction ratios, and at the same time, it outperformed systems built with all the original features and systems that use the widespread PCA-P to reduce the features. Unfortunately, test results were much worse mainly due to the small size of the training partitions. Nevertheless, similar RMSE improvements as in cross-validation could be expected in test whenever a representative training partition is provided.

One additional advantage of the proposed QE methodology is that it dramatically reduced operating times. Hence, we could take advantage of this efficiency to predict translation quality from hundreds of features. Results have shown that PLS-P was able to efficiently manage more than a thousand features to largely improve prediction accuracy. Alternatively, this time-efficiency makes this approach well-suited to be deployed in scenarios with strict temporal restrictions, such as the interactive MT framework in the next chapter.

Chapter

4 Active Interaction for Interactive MT

Current MT technology is far from perfect. To be successfully embedded in real-world applications, it must compensate for its imperfections by interacting intelligently with the user. The *interactive machine translation* (IMT) paradigm, where both an statistical MT model and a human expert collaborate to generate the translation, has been shown to be an effective computer-assisted translation approach. However, in the conventional IMT approach, the user is assumed to systematically supervise each successive translation generated by the system and find the point where the next translation error appears. From the system's point of view this is a *passive* protocol since the system just waits for the human feedback, without concern about how the supervision decisions are made. Here, we propose an alternative *active* protocol where the system informs the user about which translation elements are worthwhile to supervise. Specifically, we propose to use quality estimation techniques to locate translation errors where user attention shoud be focused.

First, we start in Section 4.1 with a brief introduction that motivates the proposed active protocol. Then, Section 4.2 provides a description of the proposed active interaction protocol for IMT. Next, Section 4.3 describes the experiments carried out to evaluate our proposal, and Section 4.4 presents the results of these experiments including an evaluation involving actual human users. We conclude with a summary of the work in Section 4.5.

Chapter Outline

4.1 Introduction

Research in the field of MT aims at developing computer systems which are able to translate between natural languages without human intervention. Unfortunately, the quality of the translations that can be generated by current state-of-the-art MT technology still remain below than that of human translation [Callison-Burch et al., 2012]. Of course, the quality level of fully-automatic translations can be enough for many applications. However, for those applications that require high-quality translations, automatic translations have to be supervised by a human expert in order to reach publishable level. This approach where human translators use MT technology to support and facilitate the translation process is known as computer assisted translation (CAT) [Isabelle and Church, 1998].

An efficient CAT approach has already been introduced in Section 1.5. The interactive machine translation [Barrachina et al., 2009] (IMT) approach uses a fully-fledged SMT system to produce translation hypotheses, or portions thereof, that can be accepted or amended by a human expert. After each user interaction, the IMT system uses user feedback to generate improved translations. This collaboration between the user and the MT system, where the system is guided by the human user and the user is assisted by the system to complete the translation, has the potential to significantly reduce the human effort required to generate the translations [Casacuberta et al., 2009].

Here, we plan to revise the IMT approach, and more precisely its interaction protocol, in order to further reduce the effort required from the user to generate the translations. In the conventional IMT approach (see Section 1.5), the user is assumed to systematically supervise each successive system translation and find the point where the next translation error appears, see Figure 1.3. In other words, the system passively responds to user feedback without any concern about which translation elements should be supervised. In contrast, we propose an alternative protocol where the system actively informs the user about which of the generated translations (or parts thereof) are likely to be incorrect, and the user then decides how to proceed. By helping the user to locate possible translation errors, the proposed active protocol has the potential to facilitate the human-system interaction, and hence, to improve the overall system-human translation performance. For instance, if a slight degradation in translation quality can be tolerated for the sake of efficiency, then the user might validate the system output after only supervising (a few) marked words. Active interaction protocols have been successfully applied in the development of efficient speech recognition [Wessel and Ney, 2005], handwriting text recog-

source (**f**): Transferir documentos explorados a otro directorio
desired translation ($\hat{\mathbf{e}}$): Move scanned documents to another folder

interaction-0	$\mathbf{e}_p$ $\mathbf{e}_s$	*Move documents scanned to other directory*
interaction-1	$\mathbf{e}_p$ k $\mathbf{e}_s$	Move **s** *canned documents to other directory*
interaction-2	$\mathbf{e}_p$ k $\mathbf{e}_s$	Move scanned documents to other **f** *older*
accept	$\mathbf{e}_p$	Move scanned documents to other folder

Figure 4.1: Example of IMT session with word-level active interaction. System suggestions are in italics, validated prefixes are printed in normal font, and user corrections are boxed. The parts of the translation considered as incorrect by the system are displayed in red. The final output differs from the desired translation $\hat{\mathbf{e}}$, but it is nevertheless a valid translation of the source sentence $\mathbf{f}$.

nition [Serrano et al., 2013], and information extraction [Kristjansson et al., 2004] systems.

Note that despite being presented within the IMT framework, active interaction is a general protocol that does not make any assumption about the procedure followed to supervise the translations. Therefore, it can be applied to other CAT approaches, e.g. the conventional post-edition approach.

The potential advantages of the proposed active interaction protocol are illustrated when we compare it (Figure 4.1) with the conventional IMT session depicted in Figure 1.3. Both translation sessions involve the translation of the same Spanish sentence "Transferir documentos explorados a otro directorio", and its goal is to obtain the same English translation "Move scanned documents to another folder". In the conventional IMT session, the user has no information about the reliability of the translated words, so he must assume that all words are equally likely to be correct or incorrect. After three interactions with the system, each one involving a prefix validation and the introduction of a correction, the user finally obtains the desired translation. With the proposed active interaction protocol, exemplified in Figure 4.1, the translation process can be quite different. At interaction zero, the system estimates that the words "documents scanned" are an incorrect translation (words considered to be incorrect are displayed in red in the example). With this in-

formation the user can focus directly on that words to correct them, skipping the word "Move". Then, the system provides a new suffix and highlights the target word "directory" as incorrect. Again, the user can directly focus in this word and correct it; with possibly only a brief look to the rest of the suggested suffix. Finally, the system considers all the words in the suggested suffix to be correct, so the user may accept it with no further consideration. Following the conventional IMT approach the user has to check the correctness of six words and introduce three corrections to obtain the desired translation, while in this example, the user has to check the correctness of only three words and introduce two corrections to obtain the final translation.

In the proposed active interaction protocol depicted in Figure 4.1, the systems suggests the user which words of the generated translation are worthy of supervision. That is, the system implements an active interaction protocol at the word-level. Alternatively, we can devise an active interaction protocol where the decisions about the need for user supervision are taken for the generated translations as a whole. In this case, the interaction process will be as follows. For each automatically generated translation, the system actively informs the user of its reliability. If the translation is good enough, the user may choose to skip the translation without further supervision. In other case, the user and the system collaborate to obtain the correct translation in a conventional IMT translation session as exemplified in Figure 1.3. As for word-level active interaction, important effort reductions are potentially achievable. In this case, a user may need to supervise only a few sentences to obtain the complete translation of a whole text corpus.

Note that with a passive protocol, "perfect" results are guaranteed from the user point of view since it is the user himself who is fully responsible of the accurateness of the translations. With an active protocol, on the other hand, the quality of the results may depend on the system ability to select appropriate elements for supervision. For example, a busy translator may be willing to only supervise those translations elements suggested by the system. By focusing user attention to those translation parts that are more likely to be erroneous, active interaction may result in better compromises between user effort and translation quality than the conventional passive protocol.

4.2 Implementation of Active Interaction for IMT

The key function required to implement the proposed active interaction protocol is the ability of the MT system to estimate the reliability of its own generated translations. Clearly, this functionality is a quality estimation (QE)

task such as that described in Chapter 3. Depending on the chosen active interaction protocol, the system has to provide quality information at word- or sentence-level. Next section describe our proposal to implement both options.

4.2.1 Word-Level Active Interaction

Word-level QE is typically addressed as a classification problem [Gandrabur and Foster, 2003; Blatz et al., 2004; Ueffing and Ney, 2007; Sanchis et al., 2007]. Given a target language sentence (and potentially other additional sources of information) generated by an MT system, a set of features is extracted for each translated word. Note that these features can be consider as individual estimators for word-level quality. Then, a model trained using a particular machine learning algorithm is employed to compute from these features the "probability" of each word of being incorrect. Different TransType-style MT systems [Foster et al., 2002; Gandrabur and Foster, 2003; Ueffing and Ney, 2005] have used quality information to improve translation prediction accuracy. Here, we propose in contrast a different application where quality information is used to aid the user to interact with the IMT system.

Several word-level QE approaches have been proposed in the literature [Gandrabur and Foster, 2003; Blatz et al., 2004; Ueffing and Ney, 2007; Sanchis et al., 2007] (see Section 3.5.3). These methods were designed to maximize prediction accuracy while considering temporal and spatial complexity of its computation as secondary issues. This approach is reasonable in classical pattern recognition scenarios where the predictions of a QE system are to be evaluated in a separate step. In our case, however, QE must be provided within an interactive environment where the user experience is crucial. Therefore, the ability to compute the quality predictions in real-time is a key characteristic to be taken into account. In other words, the QE method chosen to implement active interaction for IMT has, of course, to be as accurate as possible, but it also has to be very fast to compute so that it does not interfere with the interaction process.

A particular consequence of the time constraints inherent to interactive environments is that they specifically limit the complexity of the possible classification models. The temporal complexity of a QE system is given by the complexity of computing the features plus the complexity of the classification model. Note that we are only interested in the complexity involved in performing the prediction, we can ignore the complexity of the training and tuning steps since they can be carried out previous to the interaction. As described in Section 3.5.3, the computation of the features usually involves a constant, or at most linear, time complexity given the input string. In contrast, the com-

plexity of computing the quality score, except for the simplest classification models, usually involves more complex calculations that account for most of the complexity of the QE system.

Taking these considerations into account, we decide to discard the use of a classification model and focus on computing a single feature as a direct estimator of the quality of the words. Given the quality estimator, we can then classify each word as "correct" or "incorrect" depending on whether its quality score excess or not a certain word classification threshold ρ_w. Note that other QE approaches, see Chapter 3, could surely provide better performance but their higher computational complexity forbids their use in an interactive environment.

Note that by varying the value of the classification threshold ρ_w we can modify the behavior of the proposed active interaction protocol. Particularly, we can range between a fully-automatic SMT approach where all words are considered as correctly translated ($\rho_w = 0.0$), and a conventional IMT approach where all words are considered as incorrectly translated, namely suitable to be supervised, ($\rho_w = 1.0$).

Regarding the particularly chosen quality estimator, we implement the word-to-word lexicon feature described in Chapter 3. Formally, given a target language sentence $\mathbf{e} = \mathrm{e}_1 \dots \mathrm{e}_i \dots \mathrm{e}_{|\mathbf{e}|}$ translation of a source language sentence $\mathbf{f} = \mathrm{f}_1 \dots \mathrm{f}_j \dots \mathrm{f}_{|\mathbf{f}|}$, we follow [Ueffing and Ney, 2005] estimating the quality, $\Upsilon(\mathrm{e}_i, \mathbf{f})$, of each target word e_i given the source sentence $\mathbf{f}$ as the maximal lexicon probability of the contribution of the word to the total probability of translation $\mathbf{e}$:

$$\Upsilon(\mathrm{e}_i, \mathbf{f}) = \max_{0 \leq j \leq |\mathbf{f}|} \mathrm{P}(\mathrm{e}_i \mid \mathrm{f}_j) \tag{4.2.1}$$

where f_0 is the "empty" or "null" word, introduced to capture a target word that corresponds to no actual source word, and $\mathrm{P}(\mathrm{e}_i \mid \mathrm{f}_j)$ is the word-to-word lexicon, namely the probability of target word e_i of being the translation of source word f_j. A detailed description of this word-level quality estimator has been provided in Section 3.5.3.

We choose this estimator because it relies only on the source sentence and the proposed translation, and not on an N-best list of translations or an additional estimation layer as many other features do [Blatz et al., 2004; Sanchis et al., 2007] (see Section 3.5.3). Thus, it can be calculated very fast during search, which, as we have said above, is crucial given the time constraints inherent to interactive systems. Moreover, its accuracy to estimate the quality is similar to that of other word-level features as the results presented in [Blatz et al., 2004; Sanchis et al., 2007] show.

4.2.2 Sentence-Level Active Interaction

As we have described in Chapter 3, sentence-level QE is usually addressed as a regression problem [Quirk, 2004; Blatz et al., 2004; Specia et al., 2009b] where multiple features are combined to predict a quality score. However, this approach is too much time-consuming to be implemented in an interactive scenario where system response must be given in real-time. Therefore, as we have done for word-level active interaction, we decide to discard the use of a regression model and compute a direct estimator of the quality of the translations.

We can estimate the quality of the translations by directly computing its Model 1 [Brown et al., 1993] score, see Equation 3.5.1. However, while the word-level quality estimator in Equation 4.2.1 have shown a quite good prediction accuracy [Blatz et al., 2004; Sanchis et al., 2007], Model 1 only provides a discrete performance in scoring full translations and has been outperformed by more complex SMT models, e.g. log-linear models [Och and Ney, 2002]. Therefore, we considered that a proper combination of the quality scores of the individual words can be a more efficient estimator of sentence-level translation quality.

Given a target sentence $\mathbf{e}$ translation of a source sentence $\mathbf{f}$, we compute a sentence-level quality estimator, $\Upsilon(\mathbf{e}, \mathbf{f})$, by combining the quality scores of the target words e_i in $\mathbf{e}$. We compute two different estimators that differ in the way the word-level quality estimations $\Upsilon(\mathrm{e}_i, \mathbf{f})$ (see Equation (4.2.1)) are combined:

Mean: Geometric mean of the word-level quality scores:

$$\Upsilon(\mathbf{e}, \mathbf{f}) = \sqrt[|\mathbf{e}|]{\prod_{i=1}^{|\mathbf{e}|} \Upsilon(\mathrm{e}_i, \mathbf{f})} \tag{4.2.2}$$

Ratio: Percentage of words in the translation classified as correct. A word is classified as correct if its quality score exceeds a certain word classification threshold ρ_w:

$$\Upsilon(\mathbf{e}, \mathbf{f}) = \frac{|\{\mathrm{e}_i \mid \Upsilon(\mathrm{e}_i, \mathbf{f}) > \rho_w\}|}{|\mathbf{e}|} \tag{4.2.3}$$

The reasons to choose these particular estimators are similar to those presented for word level active interaction. They can be computed very fast directly from the source sentence and the proposed translation. Therefore, they

can be applied to a wider range of MT systems than other direct estimators of sentence-level translation quality. For instance, an estimator based on posterior probabilities would require the MT system to be able to generate lists of N-best translations.

Finally, each translation is classified as erroneous or not depending on a sentence classification threshold ρ_s. Again, the value of the threshold allows us to adapt the system to the requirements of each particular task ranging between a fully-automatic SMT system ($\rho_s = 0.0$, no translation is suggested for user supervision) and a fully-supervised IMT system ($\rho_s = 1.0$, all translations are suggested for user supervision). Alternatively, we can implement a binary classifier that uses the above described estimators as features to classify the translations. However, this additional estimation layer would also increase computation time, reason why we choose to implement the simpler threshold-based classification.

4.3 Experimental Setup

Next, we describe how the empirical evaluation of the proposed active interaction protocol was carried out. Specifically, we describe the used corpus, the experimental methodology with a simulation of the user, and the assessment measures implemented to evaluate performance.

4.3.1 Corpus and Methodology

The active interaction protocols introduced in the previous section were assessed through a series of IMT sessions involving a simulated user. We used the EU corpora [Khadivi and Goutte, 2003] to perform the experiments. The three bilingual EU corpora were extracted from the *Bulletin of the European Union*, which exists in all official languages of the European Union and is publicly available on the Internet[a]. These corpora were acquired and processed in the framework of the TransType2 project and have been previously used to evaluate IMT approaches [Barrachina et al., 2009]. Specifically, we carried out the experiments on the Spanish–English corpus of the EU corpora. This corpus was divided intro three separate sets: one for training, one for development, and one for test. The main figures of the corpus are displayed in Table 4.1.

For our experiments we used an IMT system based on a log-linear SMT model. We built a log-linear MT system (see Section 1.3.3) using the Thot [Ortiz-Martínez et al., 2005] toolkit. The weights of the log-linear were tuned by

[a] http://ec.europa.eu/archives/bulletin/en/welcome.htm

		Spanish	English
Training	Sentences	214.5K	
	Running words	5.8M	5.2M
	Vocabulary	97.4K	83.7K
Development	Sentences	400	
	Running words	11.5K	10.1K
	Perplexity (trigrams)	46.1	59.4
Test	Sentences	800	
	Running words	22.6K	19.9K
	Perplexity (trigrams)	45.2	60.8

Table 4.1: Main figures of the Spanish–English corpus of the EU corpora. K and M stand for thousands and millions of elements respectively.

MERT [Och, 2003], optimizing the BLEU score on the development partition. To efficiently compute the suffixes, we followed the standard search implementation described in [Barrachina et al., 2009]. Such implementation is based on the use of word-graphs [Ueffing et al., 2002]. We chose Thot over the widespread Moses [Koehn et al., 2007] toolkit because preliminary experiments revealed that the word-graphs generated by Thot report a better performance than the ones by Moses when used in IMT. In addition, fully-automatic translation experiments revealed no statistically significant difference in translation quality between Thot and Moses. Finally, the decoder was set to only consider monotonic translations because non-monotonic translations generate huge word-graphs that result in an excessive response time for the suffix search.

We used this trained SMT model to implement the active interaction protocols for IMT described in the previous section. A probabilistic word lexicon was also estimated using the training partition of the corpus. This lexicon was used in the experiments to compute the quality score of each translated word (Equation (4.2.1)) or sentence (Equations (4.2.2) and (4.2.3)). Given the quality scores, each word or sentence suggested by the system was marked as a possible error if its score is below a given classification threshold. Finally, a simulated user was in charge of supervising the suggested translations.

4.3.2 User Simulations

The straightforward evaluation of the proposed active interaction protocol for IMT would involve human experts. However, such a user study where a user is asked to translate hundreds of sentences is very costly both in terms of resources and time. Therefore, instead of a human evaluation, we carried out an evaluation using a simulation of the potential human users. To do that, we followed previous works on IMT [Barrachina et al., 2009] and considered the reference translations as the translations a human user would want to obtain.

User Simulation for Word-Level Active Interaction

To simulate the word-level active interaction exemplified in Figure 4.1, we modify the user model typically used in IMT [Barrachina et al., 2009]. Instead of searching for the first word in the suffix that differs from the reference (see Section 1.5 and Figure 1.3), we simulate a user that absolutely relies on the quality information when interacting with the system. In other words, the decision on which parts of the sentence are to be supervised is taken based solely on the quality estimates provided by the system.

By adopting such a user model, we are making the two quite strong assumptions. On the one hand, we are assuming that the QE model has a perfect accuracy, i.e. we assume that it does not incur in classification errors, so the user has to supervise only those words that are classified as incorrect. Of course, QE is not perfect and some words may be misclassified. This fact, together with the ambiguity inherent to natural language, allows us to affirm that the output generated by our user simulation will not always be equal to the reference translation. On the other hand, we are assuming that the user is always able to effectively correct a word without taking into account the context of this word. This assumption is a consequence of the first one: if we skip words that may be incorrect, the user has to be capable of correcting a translated word even when the context of the word may be erroneous.

Finally, we must specify how the simulated user corrects, if necessary, the word being supervised. To do that, we use the reference translations. The idea is to detect which word in the reference corresponds to the word being supervised and, if both differ, to change the supervised word by the reference word. There are several options to compute this correspondence [Ueffing et al., 2003; Sanchis et al., 2007], but since preliminary experiments showed that all of them led to similar results, we chose the simpler approach of aligning each supervised word with the word in the same position in the reference translation.

We are aware that the proposed user simulation may seem unrealistic, but it has been developed to magnify the impact of the proposed active interaction protocol, so that its influence becomes easier to evaluate. In other words, our user simulation is not designed to imitate the behavior of an actual IMT user, but it aims at measuring to what extent the information provided by the proposed active interaction protocol may facilitate the human interaction with the IMT system.

User Simulation for Sentence-Level Active Interaction

We follow the same ideas described above to define a user model for the active interaction protocol at the sentence-level. To take the supervision decisions, our simulated user again relies blindly in the quality information provided by the system. In this case, instead of supervising a few words (the ones classified as incorrect) of each sentence, the user will supervise only a few sentences of each text corpus.

Obviously, the same two assumptions described for word-level active interaction are also made here. However, while we still assume that quality information is "perfect" which is a quite strong assumption, the assumption that the user is able to correct a translation without taking into account the context is now more realistic. Certainly, knowing the precedent or the following sentences to be translated may provide valuable information to efficiently translate a given sentence. However, it is also clear that the meaning of a sentence is usually self-contained which allows the user to generate reliable translations even in the absence of context.

Finally, the correction of the translation being supervised can be done straightforwardly by implementing any CAT supervision approach. In our experiments, we followed the conventional IMT approach depicted in Figure 1.3.

4.3.3 Assessment Measures

The evaluation of our proposal was threefold. First, we wanted to measure the classification accuracy of the system. To do that, we computed the *classification error rate* (CER) which is defined as the number of misclassified elements divided by the total number of elements. Second, we wanted to measure the effort the user have to invest to obtain the translations. Our user simulation works on a word-level, thus we evaluated the effort of the simulated user by means of the word stroke ratio [Tomás and Casacuberta, 2006] (WSR) described in Section 1.6.2. Lastly, since the generated translations may be different from the reference, we also measured how much they differed us-

ing the BLEU [Papineni et al., 2002] and TER [Snover et al., 2006] measures described in Section 1.6.1.

4.4 Experiments

We now describe a series of experiments carried out to study the proposed active interaction protocol for IMT. Specifically, we wanted to measure to which extent the quality information provided by the system may help the user to interact with the system. Unfortunately, no adequate mathematical tools are available to measure how friendly and effective an interaction protocol is, and only intuition and trial-and-error can be used generally.

According to these considerations, we divided the experimentation into two separate series. In a first block of experiments, we performed an "in-laboratory" study of the proposed active interaction protocol. We were focused in obtaining quantitative measurements that allow us to compare our approach to the conventional IMT based on objective properties. Section 4.4.1 and Section 4.4.2 describe these experiments for word-level and sentence-level respectively. The second block of experiments involved different translation sessions performed by actual human users. The objective of these experiments was in this case to collect the qualitative opinions of the users about the proposed approach. Particularly, we wanted to study how comfortable and efficient the users consider the active interaction protocol in comparison to the conventional passive IMT protocol. The results of this usability experimentation are presented in Section 4.4.3.

4.4.1 In-Laboratory Experiments for Word-Level Active Interaction

Word Classification Results

We carried out an initial experimentation intended to evaluate the chosen word-level quality estimator as a word classifier. To do that, we compared the predictions of the estimator to a reference labeling of the words. Automatic word labeling is usually a complex problem in MT for which different correction criteria have been proposed [Sanchis et al., 2007]. In our case, we aim at detecting which words are to be corrected during a conventional IMT session. Therefore, the words corrected by the user should be labeled as "incorrect" while the rest of words should be considered "correct". To do that, we carried out a conventional IMT session (considering the reference translations as the

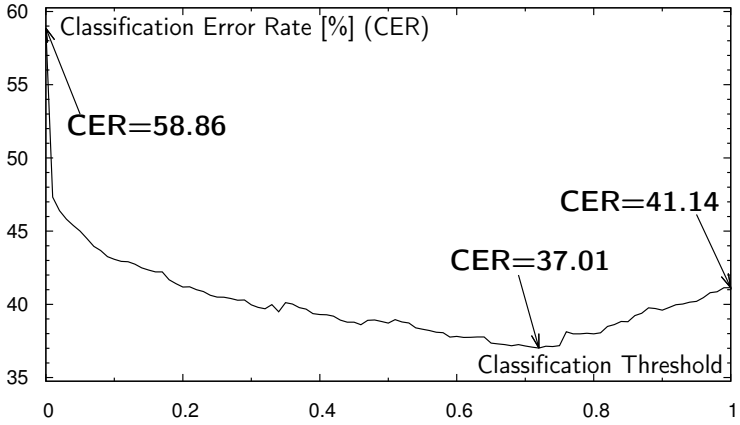


Figure 4.2: CER, as a function of the classification threshold, of the QE used to implement active interaction.

translations an actual user may want to obtain). Then, we used the interactions with the system to label the translated words. For example, in the IMT session in Figure 1.3, at iteration 1, word "Move" is labeled as "correct" because the user marked it as a valid prefix while word "documents" is labeled as "incorrect" because the user presses key "s" to correct it. This process continues until the user accepts the translation. As a result, we obtain a reference label "correct" or "incorrect" for each word in the suffixes suggested during the IMT session. Then, we compute their quality estimations (Equation (4.2.1)) and classify each word according to a certain classification threshold. Finally, we can evaluate the classification accuracy of the quality estimator by comparing the predicted labels with the reference labels computed from the IMT interaction.

Figure 4.2 displays the classification accuracy (CER) obtained for different values of the classification threshold ρ_w. Note that for the two extreme values the quality estimation does not help to distinguish between the correct and incorrect words in the suffix: $\rho_w = 0.0$ always classifies the target words as "correct", whereas $\rho_w = 1.0$ always classifies the target words as "incorrect". According to the results in the figure, best CER score was obtained for a threshold value $\rho_w = 0.75$. That is, quality information allowed to reduce word-correctness ambiguity. Therefore, we conclude that in comparison to a conventional IMT scenario, the proposed active interaction protocol provides the user with better information to detect potential errors in the suggested translations. In other words, our active interaction protocol has the potential

to facilitate the user interaction with the system.

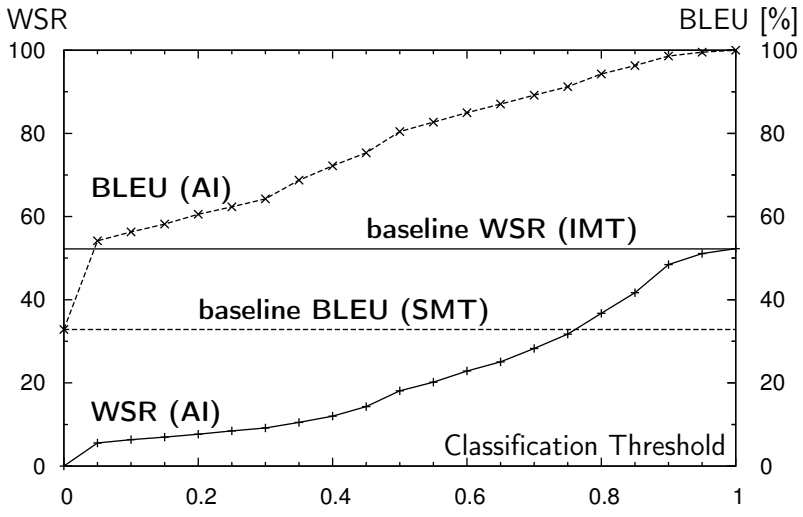
User Simulation Results

The results in the previous section indicate that the proposed quality estimator is a useful source of information to detect incorrectly translated words. Next, we aimed at evaluation which is the influence of the proposed active interaction protocol in the translation productivity of the system. Specifically, we studied the trade-off between the effort required to generate the translations and the final quality of them.

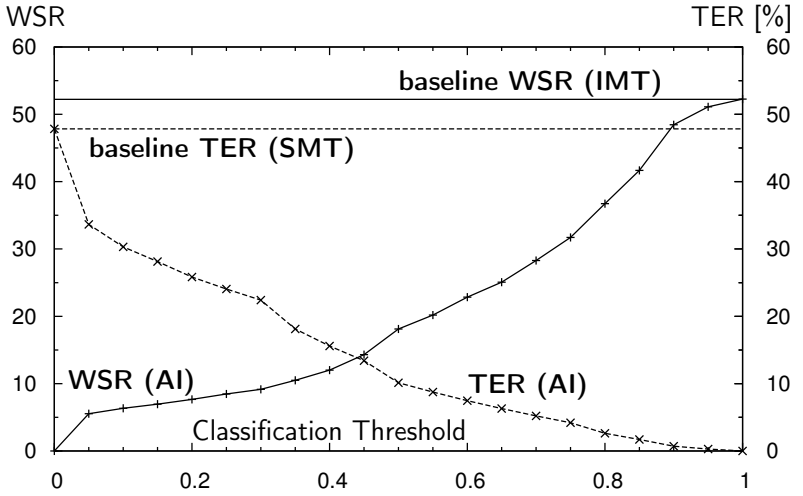
Figure 4.3 displays the results obtained by the simulated user with the proposed word-level active interaction (AI) protocol. Since the value of the classification threshold ρ_w governs the amount of words classified as incorrect, and thus the number of words supervised by the simulated user, we carried out a series of experiments ranging the value of the classification threshold ρ_w. Threshold values ranged from $\rho_w = 0.0$ (no supervision at all, equivalent to a fully-automatic SMT system) to $\rho_w = 1.0$ (full supervision, equivalent to a conventional IMT system). For each threshold value, we computed the supervision effort invested by our simulated user in terms of WSR, and the translation quality of the translations generated as a result of such supervision process. Figure 4.3(a) represents translation quality as measured by BLEU (left vertical axis) and WSR (right vertical axis) as functions of the threshold value. Additionally, we also display the baseline BLEU obtained by a fully-automatic SMT system that required no user effort, and the baseline WSR required by a conventional IMT system that always generates error-free translations. Figure 4.3(b) represents the same information but translation quality is measured by TER. Baseline systems were built with the same partitions used to build the active interaction system used by the simulated user.

Results displayed in the figure show a smooth transition between a behavior equivalent to that of a fully-automatic SMT system and the behavior of a conventional IMT system. As we raised the threshold, more words were marked as incorrect by the system, and consequently more words are supervised by our simulated user. Therefore, the proposed active interaction protocol can be seen as a generalization of the IMT approach that allows the user to adapt the system to match the requirements (amount of user effort or quality of the generated translations) of a particular translation task.

As an example we set the system to use the optimal threshold ($\rho = 0.75$) obtained in the previous word classification experiments, see Figure 4.2. Using this configuration we were able to generate almost error-free translations ($\sim$90% BLEU and $\sim$ 5% TER) by correcting only $\sim$ 30% absolute of the



(a)



(b)

Figure 4.3: User effort (WSR) and translation quality (BLEU 4.3(a) and TER 4.3(b)) as functions of the classification threshold. Word-Level active interaction (AI) was carried out by the simulated user previously described. The BLEU baseline is the BLEU score of a fully-automatic SMT system while the effort baseline is the WSR required by a conventional IMT system.

translated words. This result represents a 40% effort reduction in comparison to the conventional IMT approach, and almost a three-fold increase in translation quality respect to the translations of a fully-automatic SMT system.

4.4.2 In-Laboratory Experiments for Sentence-Level Active Interaction

Now, we describe the results obtained with the sentence-level user simulation described in Section 4.3.2. As for the word-level user simulation experiments above, the aim of this experimentation is to evaluate the influence of the proposed active interaction protocol in the supervision process, and to estimate the potential improvements in translation productivity that can be achieved.

Figure 4.4 displays the results obtained by the simulated user with the proposed sentence-level active interaction (AI) protocol. We carried out experiments ranging the value of the sentence classification threshold ρ_s between $\rho_s = 0.0$ (no supervision at all, equivalent to a fully-automatic SMT system) and $\rho_s = 1.0$ (full supervision, equivalent to a conventional IMT system). For each threshold value, we computed the effort employed by the simulated user to generate the translations in terms of WSR and the final quality of the translations in terms of BLEU. Figure 4.4(a) displays the results obtained using the `Mean` quality estimator in Equation (4.2.2), and Figure 4.4(b) displays the results for the `Ratio` estimator in Equation (4.2.3). Additionally, we display the BLEU score obtained by a fully-automatic SMT system as a translation quality baseline, and the WSR score obtained by a conventional IMT system as a user effort baseline.

Results for the `Mean` quality estimator showed a smooth transition between the behavior of a fully-automatic SMT system and that of a fully-supervised IMT system. This transition occurred between $\rho_s = 0.0$ and $\rho_s = 0.6$.

Regarding the `Ratio` quality estimator, its computation depend on a word classification threshold ρ_w. Therefore, we performed preliminary experiments ranging the value of ρ_w. The results of these preliminary experiments showed that the value of ρ_w controls the width of the interval in which the transition between an SMT system and an IMT system occurs. We selected $\rho_w = 0.4$ since it was the threshold value that resulted in a smoother transition. Figure 4.4(b) presents the WSR and BLEU scores obtained by the `Ratio` quality estimator for different values of the sentence classification threshold ρ_s. For example, with a threshold value $\rho_s = 0.6$ we could obtain almost perfect translations ($\sim$90 BLEU points) with a WSR reduction of 20% respect to a conventional IMT system. Moreover, the final translations were compared with only one reference, thus the reported quality scores are clearly pessimistic. Better results

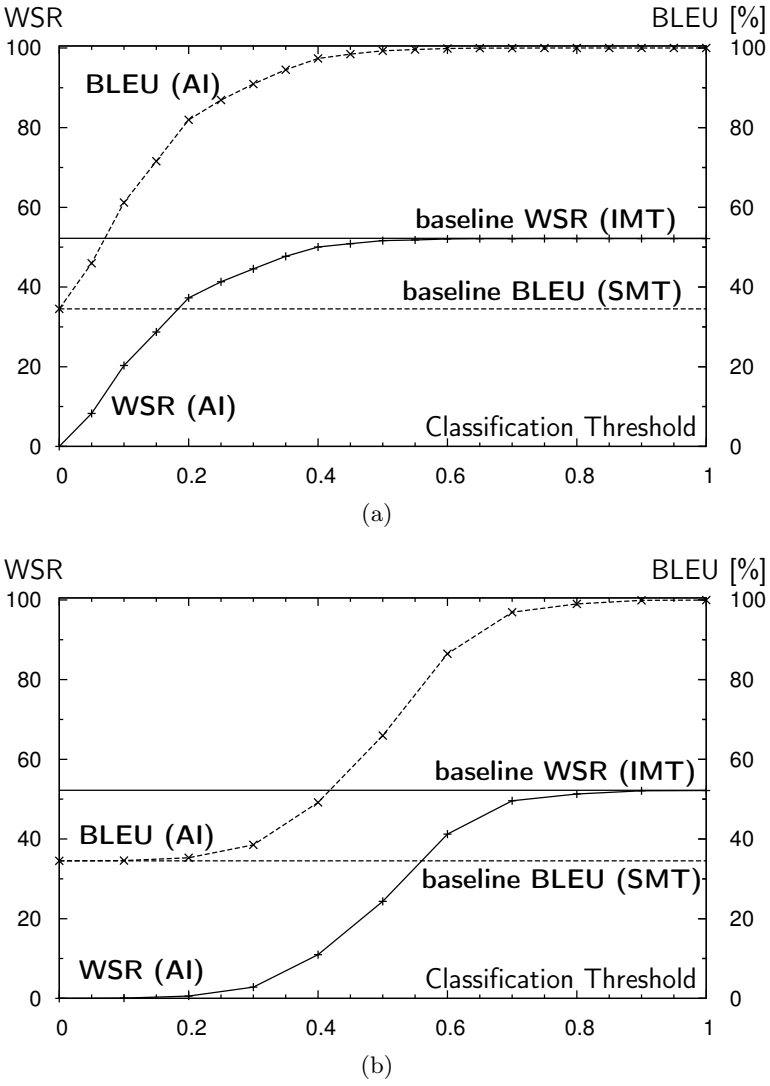


Figure 4.4: User effort (WSR) and translation quality (BLEU) as functions of the classification threshold. Results for the `Mean` quality estimator are displayed in sub-figure 4.4(a), while results for the `Ratio` quality estimator are shown in sub-figure 4.4(b). Sentence-level active interaction (AI) was carried out by the described simulated user. The BLEU baseline is the BLEU score of a fully-automatic SMT system while the effort baseline is the WSR required by a conventional IMT system.

src-1	DECLARACIÓN (No 17) relativa al derecho de acceso a la información
ref-1	DECLARATION (No 17) on the right of access to information
tra-1	DECLARATION (No 17) on the right of access to information
src-2	Conclusiones del Consejo sobre el comercio electrónico y los impuestos indirectos.
ref-2	Council conclusions on electronic commerce and indirect taxation.
tra-2	Council conclusions on e-commerce and indirect taxation.
src-3	participación de los países candidatos en los programas comunitarios.
ref-3	participation of the applicant countries in Community programmes.
tra-3	countries' involvement in Community programmes.

Table 4.2: Examples of automatic translations classified as correct, and thus, not supervised by the simulated user.

can be expected if a multi-reference corpus is used.

Finally, Table 4.2 shows the source language sentence (**scr**), the reference translation (**ref**), and the automatically generated translation (**tra**) for three translations classified as correct by the sentence-level quality estimator ($\rho_w = 0.4, \rho_s = 0.6$). Therefore, these automatic translations were not supervised by the simulated user. The first translation (**tra-1**) is identical to the corresponding reference translation (**ref-1**). The second translation (**tra-2**) is different from the reference translation (**ref-2**) but it is still a correct translation of the source sentence (**src-2**). Lastly, the third translation (**tra-3**) is an example of a slightly incorrect translation.

4.4.3 Experiments with Actual Human Translators

Aims

The experiments in the previous section were focused on translation productivity issues correlating user effort and translation quality expectation. Now, we describe the results of a series of experiments that studied the proposed active interaction protocol from a more "social" point of view. The overall aim of the experiments in this section is to analyze the impact of word-level active interaction in the supervision process performed by actual human translators. We specifically chose word-level active interaction because it actually modifies the information available to the user during the IMT session[b]. Hence, we hope

[b] In contrast, sentence-level active interaction informs the user about the quality of the full translations which, if necessary, are supervised in a conventional IMT session.

that this investigation will also supply the evidence needed to develop more friendly, effective, and efficient supervision protocols.

Field Trial Description

We report the results of an unofficial field trial carried out in the framework of the CasMaCat [CASMACAT, 2011] project. The field trial was carried out in the Copenhagen Business School using the prototype developed by the author of this thesis and other members of the Pattern Recognition and Human Technology group at the Universitat Politècnica de València. The author was also present in Copenhagen during the field trial to oversee its progress. A group of five users (three females and two males) aged between 21 and 49 volunteered to perform the evaluation. The participants were not professional translators. However, they have strong skills on the translation between the source (English) and the target (Spanish) languages. No previous domain knowledge on the topics of the texts being translated was required. When asked about previous experience in post-editing of MT outputs, 40% of them claimed to have previous experience in post-editing assignments. This difference in post-editing experience was not considered a bias in the sample of the study, since the aim was not to measure productivity but user satisfaction.

Each participant was asked to translate two blocks of text, one using a conventional IMT system and the other using the proposed word-level active interaction protocol. For each user, we randomly selected which approach to use first. Both approaches were implemented in the CasMaCat prototype. The graphical user interface of the prototype can be seen in Figure 4.5. As we can see in the figure, those words considered by the system as suitable to be supervised by the user are highlighted in different colors. Quality information on its own cannot be useful to human users because it usually does not have a direct interpretation. To made quality information easily interpretable to human users, we highlight the translated words in different colors according to two different classification thresholds. On the one hand, we highlight in red color those words that almost surely are erroneous and thus have to be supervised by the user. To do that, we searched for a threshold value that provided a high precision ($\sim$80%) in recognizing incorrectly translated words. On the other hand, we highlight in orange color those words that are dubious. We found them by using a classification threshold that provided high recall ($\sim$80%). The conventional IMT system used the same interface but no words were highlighted.

Figure 4.5: User interface of the CasMaCat prototype.

Methodology

To evaluate the influence of the word-level active interaction protocol, we focus on the feedback provided by the translators in individual interviews carried out as a final step of the field trial. These interviews were held so translators could provide feedback on their experience while working with the prototype, as well as to suggest new functions for future versions of the prototype. The interviewer used a standardized interview schedule with a set of predefined question which were asked to all respondents. The questions tended to be asked in a similar order and format to make a form of comparison between all possible answers. However, the questions, rather than restricting the answers to specific types of information, were intended to guide the discussion to relevant resources of information. There was also scope for pursuing novel, relevant information. The interviewer frequently had to formulate impromptu questions in order to follow up leads that emerged during the interview. The interviews were recorded and notes were taken of the key points made by the users.

Findings

Prior to any comparison question between the conventional IMT approach and the proposed active interaction protocol, we asked the users a few questions to evaluate to which extent active interaction have matched their expectations. Clearly, these are qualitative aspects that cannot be expressed by a quantity or a measured value. Thus, the users were asked to provide a yes / no response to each question. Of course, as we have described before, we also allow the users to clarify their answers whenever necessary. The users' responses were as follows:

	Yes	No
• Do you consider active interaction to be a desirable feature?	40%	60%
• Do you consider the provided quality information to be accurate?	0%	100%
• Do you consider the active interaction protocol to be annoying?	80%	20%

As appears from the answers and the corresponding clarifications of the users, active interaction is a feature that an important percentage of them would like to have in a potential translation workbench. However, the users were quite disappointed by the apparently poor performance of the quality information provided. This perception was what made the system annoying for the users. As stated by one participant:

> "Many times the words marked by the system as wrong were actually correct, while wrong translations remained in black. In the end I had to double-check most of the sentences to make sure that words marked in black were actually acceptable translations"

This was a quite surprising result given the good performance previously reported for the chosen quality estimator in laboratory experiments [Blatz et al., 2004; Sanchis et al., 2007]. The clarifications made by the users revealed that the main problem stems in the tendency of the system to classify as incorrect words that from the user point of view are clearly correct. For example, proper names are usually classified as incorrect since they tend to appear few times, if any, in the training data. Such errors are infrequent, so they do not penalize much the performance of the estimator as measured in CER or other automatic measures. However, these errors are quite annoying for the users who then distrust the provided quality information.

Regarding the comparison between active interaction (AI) and the conventional IMT interaction, it was mainly referred to usability aspects such as the potential difference in translation productivity between the two approaches. Again, these are qualitative aspects for which a yes / no answer, plus a possible clarification, was asked. Next, we provide a summary of the responses given by the users:

	IMT	AI
• Which approach do you consider to be more user-friendly?	40%	60%
• Which approach do you consider to be more useful?	60%	40%
• Which approach do you consider to be more productive?	60%	40%

From the answers of the users we can infer that they considered active interaction as an interesting protocol that has the potential to improve the usability of the conventional IMT approach. However, some users did not consider that the active interaction protocol could improve the usefulness nor the productivity of IMT systems. Again, the reason for this apparent mismatch in the users' opinions stemmed in the poor accuracy perceived of the quality estimations provided. Nevertheless, the users reckoned that active interaction has a great potential to be explored and that it would be a much desirable characteristic whenever appropriate quality information is provided. Quoting one of the participants:

> "I could definitely benefit from this type of visual aid, but the system still needs to make better predictions"

Finally, we would like to discuss the opinions of the users regarding the chosen way to display quality information within the prototype. As we have described before, we highlight some of the translated words according to two different criteria: words with a high probability of being incorrectly translated and almost surely must be corrected by the user (red color), and dubious words that should be checked by the user but do not necessary have to be wrong (orange color). The users agreed that the specific selection of colors was adequate allowing for an easy identification of the different word types. However, their opinions were mixed regarding the usefulness of the different criteria. Some considered that identifying incorrectly translated words has to be the priority, some others considered that detecting dubious parts of the translation has more interest, and other users even consider that both criteria are equally useful, and thus, both of them should be displayed. As a consensus, we conclude that both criteria must be computed but it should be up to the user to decide which of them, or both, to use to highlight the words.

4.5 Summary

We have proposed a novel active interaction protocol to substitute the passive protocol implemented by conventional IMT systems. The proposed active interaction protocol was implemented by means of a quality estimator that computes the reliability of the full generated translations (sentence-level active interaction) or words thereof (word-level active interaction). Using a variable classification threshold, the system is able to inform the user of which translation elements are likely to require supervision. We then have described an efficient implementation of this idea that addresses the practical challenges derived from the strict time constraints inherent to an interactive environment such as IMT. Finally, we have described the experimentation carried out to assess the potential advantages of the proposed interaction protocol.

A first set of experiments involved an "in-laboratory" study to analyze the proposed word-level and sentence-level active interaction protocols. Regarding word-level active interaction, we first measured the accuracy of the proposed quality estimator in classifying translated words. Results showed that the proposed estimator was able to classify the words with less errors than considering all words erroneous (as conventional IMT systems do) or than considering all words correct (as fully-automatic SMT systems do). We thus conclude that the proposed word-level active interaction protocol provides useful information that may aid the user in localizing translation errors. Then, we studied the potential influence of providing a user with this word-level quality information. To do that, we defined a simulated user that uses the reference translations of the corpus to automatically test our proposal. The results of this experiment showed that large reductions in user effort can be achieved ($\sim 40\%$ respect to a conventional IMT system) while generating almost error-free translations ($\sim$90% BLEU and $\sim$5% TER). Similar results were obtained for the proposed sentence-level active interaction protocol showed similar results. For both proposals, the value of the classification threshold governs the behavior of the system between that of a fully-automatic SMT system and that of a fully-supervised IMT system. This feature allows us to adapt our system to match the particular requirements of each translation task.

The second set of experiments was concerned about how to effectively integrate the proposed word-level active interaction protocol within an actual IMT prototype. Specifically, we wanted to measure to which extent actual human users can benefit from the quality information provided, and which are the factors that maximize this benefit. Results showed that users regard active interaction as a desirable feature that should be integrated in a potential

translation workbench. However, the quality estimation used in the current implementation was perceived as too error-prone which annoys the users and penalized the usability of the system. A common complaint of the users was that the system systematically classifies as incorrect the proper nouns, which is reasonable given that these names are usually out-of-vocabulary words. Nevertheless, users reckoned word-level active interaction as a promising approach if a more reliable confidence information were to be deployed.

Future developments in this direction will include the use of named-entity recognition systems to detect proper names, locations, or quantities and provide specialized quality scores for them, the implementation of additional direct estimators of translation quality, and the study of different fast-to-compute schemes to combine different quality estimators.

Chapter

5 Active Learning for Interactive MT

Despite being an efficient computer-assisted translation approach, the conventional interactive machine translation (IMT) technology present some flaws that, in our opinion, prevent IMT from showing its full potential. In the previous chapter, we have presented an active interaction protocol that provides for better compromises between overall user effort and final translation quality. Now, we extend these ideas into an active learning scenario where the IMT system not only estimates for which translations it may pay off to ask for user supervision, but it additionally updates its underlying translation model with user feedback after system deployment. Finally, the goal of the proposed active learning scenario for IMT is to reduce as much as possible the supervision effort required from the user to generate high-quality translations.

The chapter is organized as follows. We start in Section 5.1 with an introduction that describes the motivation of the proposed approach. Then, Section 5.2 provides a description of the proposed cost-sensitive active learning approach for IMT. Next, Section 5.3 describes the experiments carried out to evaluate the proposed approach. Finally, we conclude with a summary of the work in Section 5.4.

Chapter Outline

5.1 Introduction

As we have described in the introduction of this thesis, phenomena such as globalization have dramatically increased the needs of translation between languages. This poses a high pressure on translation agencies that must decide how to invest their limited resources (budget, manpower, time, etc.) to generate translations of the maximum quality in the most efficient way. IMT technology, where a fully-fledged SMT system collaborate with a professional translator to generate the translations, represents an efficient approach to generate the high quality translations required by translation agencies. However, despite its success, we consider that conventional IMT technology presents some flaws for which we think there is room for improvement.

Let us consider a translation agency that is continually receiving request for translation. As any other company, this translation agency wants to earn as much money as possible which implies to fulfill as many translation requests as possible. Unfortunately, the agency also has a limited amount of resources, for instance money, manpower, or time, to fulfill those requests for translation. Therefore, the agency has to efficiently use the available resources to obtain the maximum productivity, that is, the maximum translation quality at the lowest possible user effort.

Given this scenario, two conclusions are clear. On the one hand, given that translation supervision is expensive, an exhaustive supervision of all translations is unfeasible. In other words, to obtain the maximum productivity with its limited resources the translation agency is forced to select intelligently those translations for which user supervision improves most translation productivity. On the other hand, it is obvious that good candidate translations are easier to supervise than bad translations. Hence, we can boost translation productivity by improving the overall quality of the translation model.

In Chapter 4, we have already studied the impact of selecting a subset of translation elements to undergo user supervision. This selective supervision have shown to provide better compromises between user effort and translation error than the exhaustive supervision implemented in conventional IMT technology. In this chapter, we further extend the active interaction protocol presented in the previous chapter into an *active learning* framework. In this framework, the IMT system also evaluates which sentences should be supervised by the user, and additionally, the underlying translation model is continually learning from user feedback to improve its future suggestions.

The proposed active learning framework for IMT shares the key ideas that led to the development of classical active learning [Angluin, 1988; Atlas et al.,

1990; Cohn et al., 1994; Lewis and Gale, 1994] by the machine learning community. A typical active learning scenario involves a learning task for which a large amount of unlabeled data, and a labeling oracle (e.g. a human annotator) are available. The active learner is allowed to ask the oracle to label the data from which it learns. The oracle is able to provide the correct label for any of the unlabeled data instances but each query involves a certain cost (e.g. human effort). Finally, the objective of active learning is to minimize the number of queries required to obtain a prediction model of a certain accuracy. Such learning framework have already been successfully applied to a number of natural language processing tasks such as sequence labelling [Settles and Craven, 2008], parsing and information extraction [Thompson et al., 1999], or machine translation [Haffari et al., 2009; Ambati et al., 2011].

Most active learning methods [Settles, 2009] consider a *pool-based* setting where the unlabeled data is fixed and known in advance. However, this is not the case of IMT which is built over the implicit assumption that the sentences to be translated behave as a text stream (see Figure 1.2). In order to apply active learning to a data-stream environment such as IMT, we have to face two main challenges [Zhu et al., 2010]:

- The unlabeled data is unbounded and dynamically changing. This implies that the data to be labeled by the user cannot be selected via exhaustive search in a finite data pool. In contrast such decisions must be taken at each instance (or small block of instances) during a single scan of the data stream.

- Because the data is unbounded, building one single model from all labeled data is impractical. Hence, we must rely on incremental learning techniques to update our models.

We propose an active learning framework for IMT that addresses these challenges and has the same elements that define any active learning system. Given a source language text, we have a set of automatically generated translations (that correspond to the unlabeled samples), and a human expert that can supervise and correct (label) them. The system is allowed to ask the user to supervise a subset of the automatic translations, and use the correct translations to update its SMT model. The user is able to supervise any translation but each translation supervision involves a certain amount of effort. Finally, our objective is to minimize the supervision effort required to generate a high-quality translation of the source text. Or alternatively, given a certain effort level to generate a translation of the source text of the highest possible quality.

The proposed active learning framework has several potential advantages over the conventional IMT technology. On the one hand, the selective supervision protocol allows us to limit the amount of effort to be invested in the translation process and, by supervising those sentences for which the investment of user effort is estimated to be more profitable, we also maximize the utility of each user interaction. On the other hand, the underlying SMT model is continually updated with the new available sentence pairs after deployment. Therefore, the SMT model is able to learn new translations and to adapt its outputs to match the preferences of the user. As a results, the following translations generated will be closer to those preferred by the user and the effort required to supervise them will be reduced. Additionally, all this sophistication is transparent to the user that can interact with the system in the same way that he does with a conventional IMT system.

An important aspect that determines the practical development of the proposed active learning framework is the interaction with the user. This interactivity imposes a strict temporal bound to the response time of the system, thus constraining the models and techniques that can be used to implement the different features of the framework. Related to the second challenge of active learning for data streams described above, these bounds are particularly restrictive for the model updating feature. Learning for SMT is usually implemented as a batch process. However, the temporal complexity of such process grows with the number of training samples making this approach impractical in our interactive scenario. To address this challenge, we implement the on-line learning process for SMT described by Ortiz-Martínez et al. [2010]. This on-line learning process is able to incrementally update the SMT models in constant time which permits the practical implementation of the proposed active learning framework for IMT.

Additionally, it should be noted that the active learning framework described above is a general technique that is independent of the particular approach chosen to supervise the translations. Therefore, despite being presented in an IMT framework, it can be straightforwardly applied to other CAT approaches such as post-edition.

5.2 Active Learning for IMT

This section describes in detail the proposed active learning framework for IMT. This framework is built on the foundations of the conventional IMT scenario [Barrachina et al., 2009] from which we import its user-machine interaction process (Figure 1.3) to efficiently supervise the individual translations.

However, to implement the active learning ideas (selective sampling and SMT model updating) in this scenario, we have to modify the conventional IMT work-flow. Specifically, the proposed work-flow (Section 5.2.1) asks the user to supervise only a subset of the automatic translations. These automatic translations are selected according to a particular ranking function (Section 5.2.2), and once supervised by the user, their correct translations are used to incrementally update the SMT system used by the IMT system (Section 5.2.3).

This active learning framework for IMT can be seen as an extension of the sentence-level active interaction protocol described in Chapter 4. However, while the goal of active interaction was to aid the user to identify possible translation errors, now we explicitly aim at generating high quality translations as effortlessly as possible. Of particular importance for active learning is the inclusion of the MT system as a dynamic element that may be improved in order to reduce user effort. As a result, the criterion followed to suggest translations for human supervision differs from the one followed by the active interaction system presented in the previous chapter.

5.2.1 Translation Work-Flow and Supervision Protocol

The work-flow of the proposed active learning framework for IMT implies two important differences respect to the work-flow of conventional IMT technology. On the one hand, the user no longer supervises all the automatic translations but only a subset of them. On the other hand, the final translations generated in collaboration with the user are used to update the underlying SMT model. As a consequence, the translations finally generated are a mixture of automatic and user-supervised translations. In other words, final translations may be different from the ones the user wants to obtain. In exchange, the SMT model is able to learn new translations and to adapt its output to the translation preferences of the user. This prevents the user from correcting repeatedly the same translation mistakes thus reducing the supervision effort required to obtain translations of high quality.

It should be noted that both automatic and user-supervised translations can be used to update the SMT model. However, preliminary experiments showed that updating the SMT model with automatic translations in addition to user-supervised translations (resembling a semi-supervised learning scenario [Blum and Mitchell, 1998; Chapelle et al., 2006]) resulted in worse performance than using only the user-supervised translations.

An interesting property of the proposed active learning approach is that we can modify the ratio of automatic translations supervised to the user according to the available manpower or the requirements of the task. For instance,

the translation agency presented in the introduction of this chapter may be willing to sacrifice some translation quality in exchange for improved productivity. Certainly, this is an unrealistic scenario in some cases, for example it is inconceivable not to fully-supervise the translation of a legal document such as a contract. However, there are many other translation tasks, e.g. manuals for electronic devices, or twitter and blog postings, that match this productivity-focused scenario.

Conventional IMT technology is built over the implicit assumption that the inbound text to be translated behave as a text stream (see Figure 1.2). Source sentences are translated separately and no information is stored (or assumed) about the preceding (or following) sentences, e.g. how many sentences remain untranslated. Since the IMT framework uses static SMT models and requires the user to supervise all translations, this is not a strong assumption. However, we have to take it into account because information about previously supervised translations, and particularly, about following sentences may help to estimate which automatic translations should be supervised by the user which, in turn, has great impact in the final user effort. We handle the unbound text stream by partitioning the data into blocks. Each block can be seen as an individual document to be translated. Within a block, all sentences are available, but once the algorithm moves to the next block, all sentences in previous blocks become inaccessible. We use the sentences within a block to estimate the current distribution of sentences in the stream, so that the estimation of the utility of supervising the translation of a sentence can be done as accurately as possible.

Algorithm 5.1 shows the pseudo-code that implements the proposed active learning scenario for IMT. The algorithm takes as input a stream of source sentences $\mathcal{D}$ to be translated, an initial SMT model $\mathbb{M}$, and an effort level ρ denoting the percentage of sentences of each block to be supervised. The algorithm starts by reading from the stream $\mathcal{D}$ the new block of sentences $\mathcal{B}$ to be translated (line 3). As we have said in above, IMT is intrinsically a stream translation problem, but we extract a block of sentences from $\mathcal{D}$ so that the utility of each sentence can be estimated as accurately as possible. Then, the system selects which of the sentences in $\mathcal{B}$, $\mathcal{S} \subseteq \mathcal{B}$, should be supervised by the human expert (line 4). This selection is commonly known as *sampling* in the active learning literature. Next, the algorithm translates each of the sentences in $\mathcal{B}$. Initially, the SMT model generates an automatic translation, $\hat{\mathbf{e}}$, for source sentence $\mathbf{f}$ (line 6). If the sentence has been sampled as worthy of supervision, $\mathbf{f} \in \mathcal{S}$, the user collaborates with the system to obtain the

Algorithm 5.1: Work-flow of the proposed active learning IMT scenario.

input : $\mathcal{D}$ (stream of source sentences)
$\mathbb{M}$ (initial SMT model)
ρ (effort level, percentage of sentences to be supervised)

output : $\mathbf{e} \in \mathcal{E}$ (translation generated for each of the source sentences in $\mathcal{D}$. These translations are a mixture of automatic and user-supervised translations.)

auxiliar : getNextBlock($\mathcal{D}$) (returns the next block of sentences from $\mathcal{D}$)
sampling($\mathbb{M}, \mathcal{B}, \rho$) (sampling strategy, returns the sentences to be supervised: ρ% of the sentences in $\mathcal{B}$)
translate($\mathbb{M}, \mathbf{f}$) (returns the automatic translation for $\mathbf{f}$ according to $\mathbb{M}$)
validatedPrefix($\mathbf{e}$) (returns the prefix validated by the user in the IMT interaction)
genSuffix($\mathbb{M}, \mathbf{f}, \mathbf{e}_p$) (returns the suffix that continues prefix $\mathbf{e}_p$)
validTranslation($\mathbf{e}$) (returns *True* if the user accepts translation $\mathbf{e}$ and *False* otherwise)
update($\mathbb{M}, (\mathbf{f}, \mathbf{e})$) (returns SMT model $\mathbb{M}$ updated with bilingual pair $(\mathbf{f}, \mathbf{e})$)

```
1  begin
2      repeat
3          B = getNextBlock(D);
4          S = sampling(M, B, ρ);
5          foreach f ∈ B do
6              ê = translate(M, f);
7              if f ∈ S then
8                  e = ê;
9                  repeat
10                     e_p = validatedPrefix(e);
11                     ê_s = genSuffix(M, f, e_p);
12                     e = e_p ê_s;
13                 until validTranslation(e);
14                 M = update(M, (f, e));
15                 output(e);
16             else
17                 output(ê);
18     until D ≠ ∅;
```

correct translation of $\mathbf{f}$ in a conventional IMT session (lines 8–13)[a]. Then, the

[a]Other CAT supervision protocols, such as post-edition, can also be used. This will imply a modification of lines 8 to 13 in Algorithm 5.1 so that they behave according to the chosen

new sentence pair $(\mathbf{f}, \mathbf{e})$ is used to update the SMT model $\mathbb{M}$ (line 14), and the human-supervised translation is returned (line 15). Otherwise, we directly return the initial automatic translation $\hat{\mathbf{e}}$ as the final translation (line 17). The output of the algorithm is a mixture of automatic and user-supervised translations of the source sentences in the text stream.

In the proposed active learning framework, we import the IMT interaction protocol (Figure 1.3) to allow the user to efficiently supervise the selected subset of translations. Functions between line 8 and line 13 denote the IMT supervision procedure:

translate$(\mathbb{M}, \mathbf{f})$**:** It returns the most probable automatic translation of $\mathbf{f}$ according to $\mathbb{M}$. If $\mathbb{M}$ is a log-linear SMT model, this function implements Equation (1.3.11).

validatedPrefix$(\mathbf{e})$**:** It denotes the actions (positioning and correction of the first error) performed by the user to amend an error on a system suggestion $\mathbf{e}$. The function returns the user-validated prefix $\mathbf{e}_p$ of translation $\mathbf{e}$, including the user correction k.

genSuffix$(\mathbb{M}, \mathbf{f}, \mathbf{e}_p)$**:** It returns the suffix $\mathbf{e}_s$ of maximum probability that extends prefix $\mathbf{e}_p$. This function implements Equation (1.5.2).

validTranslation$(\mathbf{e})$**:** It denotes the user decision of whether system suggestion $\mathbf{e}$ is a correct translation or not. It returns *True* if the user considers $\mathbf{e}$ to be correct and *False* otherwise.

In addition to the supervision procedure, the two elements that define the performance of Algorithm 5.1 are the sampling strategy sampling$(\mathbb{M}, \mathcal{B}, \rho)$ and the SMT model update function update$(\mathbb{M}, (\mathbf{f}, \mathbf{e}))$. The sampling strategy decides which sentences of $\mathcal{B}$ should be supervised by the user. This is a key component of our active learning framework, and has a major impact in the final performance of the algorithm. Section 5.2.2 formally defines what we mean by *"should be supervised"* and describes several strategies to measure such utility. In turn, the update$(\mathbb{M}, (\mathbf{f}, \mathbf{e}))$ function updates the SMT model $\mathbb{M}$ with a new training pair $(\mathbf{f}, \mathbf{e})$. Section 5.2.3 describes our implementation of this functionality.

5.2.2 Sentence Sampling Strategies

The goal of our active learning framework for IMT is to generate high-quality translations as effortlessly as possible. Since good automatic translations re-

supervision protocol.

quire less supervision effort that bad ones, the aim of a sampling strategy should be to select those sentences for which knowing their correct translation allows to improve most the future performance of the SMT model and require the least user supervision effort possible.

Statistical decision theory is an appealing framework to formalize this active learning problem since it offers a systematic way to represent effort-benefit trade-offs [Donmez and Carbonell, 2008]. Specifically, we use the Value of Information (VOI) [Kapoor et al., 2007] framework. This is a general approach that have been applied to a number of conventional machine learning problems where the number of classes is relatively small, e.g. image classification [Joshi et al., 2012]. In our case however the number of classes (all possible sentences in the target language) is potentially infinite. Therefore, a direct implementation of the VOI approach would be very difficult (if not unfeasible) from the computational point of view. Nevertheless, the VOI framework provides us with a solid background from where to derive computationally-efficient active learning approaches for MT.

The broad idea of the VOI framework is to select sentences based on an objective function that combines the expected quality of the future translations and the user effort required to supervise the automatic translations. In our case, from each block $\mathcal{B}$ of sentences to be translated, we have to select a subset of sentences $\mathcal{S} \subseteq \mathcal{B}$ whose automatic translations are to be supervised by the user. The objective here is to ask the user to supervise those sentences that are likely to lead to an improvement in future translation quality of the SMT model.

Given a real-valued quality function $\mathrm{Q}(\hat{\mathbf{e}}, \mathbf{e}')$ that takes values between zero and one (e.g. BLEU), the expected quality of the automatic translation $\hat{\mathbf{e}}_{\mathcal{L}}^{(\mathbf{f})} = \mathrm{translate}(\mathbb{M}_{\mathcal{L}}, \mathbf{f})$ of sentence $\mathbf{f}$ according to SMT model $\mathbb{M}_{\mathcal{L}}$ is given by:

$$\mathrm{G}_{\mathcal{L}}^{(\mathbf{f})} = \sum_{\mathbf{e}'} \mathrm{Q}(\hat{\mathbf{e}}_{\mathcal{L}}^{(\mathbf{f})}, \mathbf{e}') \cdot \mathrm{P}_{\mathcal{L}}(\mathbf{e}' \mid \mathbf{f}) \tag{5.2.1}$$

where $\mathcal{E}$ is the target language, and $\mathcal{L}$ is the set of parallel sentences used to estimate the probability distribution over translations, i.e. the set of parallel sentences used to train the SMT model. The total expected quality of the automatic translations $\hat{\mathbf{e}}_{\mathcal{L}}^{(\mathbf{f}')}$ of the sentences $\mathbf{f}' \in \mathcal{B}$ in the block is given by:

$$\mathrm{G}_{\mathcal{L}} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{f}' \in \mathcal{B}} \sum_{\mathbf{e}' \in \mathcal{E}} \mathrm{Q}(\hat{\mathbf{e}}_{\mathcal{L}}^{(\mathbf{f}')}, \mathbf{e}') \cdot \mathrm{P}_{\mathcal{L}}(\mathbf{e}' \mid \mathbf{f}') \tag{5.2.2}$$

Now, if a source sentence $\mathbf{f} \in \mathcal{B}$ is added to the set of parallel sentences by acquiring its real translation $\mathbf{e}$ from the user, the expected improvement in

quality for the translations of the sentences in $\mathcal{B}$ can be computed as:

$$\mathrm{G}_{\mathcal{L}'} - \mathrm{G}_{\mathcal{L}} = \frac{1 + \sum_{\mathbf{f}' \in \mathcal{B} \setminus \{\mathbf{f}\}} \sum_{\mathbf{e}'} \mathrm{Q}(\hat{\mathbf{e}}_{\mathcal{L}'}^{(\mathbf{f}')}, \mathbf{e}') \cdot \mathrm{P}_{\mathcal{L}'}(\mathbf{e}' \mid \mathbf{f}')}{|\mathcal{B}|} - \frac{\sum_{\mathbf{f}' \in \mathcal{B}} \sum_{\mathbf{e}'} \mathrm{Q}(\hat{\mathbf{e}}_{\mathcal{L}}^{(\mathbf{f}')}, \mathbf{e}') \cdot \mathrm{P}_{\mathcal{L}}(\mathbf{e}' \mid \mathbf{f}')}{|\mathcal{B}|} \tag{5.2.3}$$

where $\mathcal{L}' = \mathcal{L} \cup \{(\mathbf{f}, \mathbf{e})\}$. Note that the expected quality score for the translation of $\mathbf{f}$ is equal to 1.0 because the user has provided us with its correct translation. The above expression captures the *value* of querying the user for the translation of $\mathbf{f}$ and adding the pair to the corpus of parallel sentences. However, we also need to consider the *effort* required from the user to provide the actual translation of $\mathbf{f}$. Let $\mathrm{E}(\mathbf{f})$ be a function that returns a positive real number that quantifies the effort required to obtain the translation of $\mathbf{f}$. In our active learning framework, we wish to actively choose the sentences that reduce the effort incurred while maximizing the improvement in expected translation quality. The joint objective that represents the value of information for a sentence $\mathbf{f}$ can be computed as:

$$\mathrm{VOI}(\mathbf{f}) = \frac{\mathrm{G}_{\mathcal{L}'} - \mathrm{G}_{\mathcal{L}}}{\mathrm{E}(\mathbf{f})} \tag{5.2.4}$$

As defined $\mathrm{VOI}(\mathbf{f})$ denotes the expected improvement in translation quality in the translations of the corpus $\mathcal{B}$ per unit of user effort. Therefore, we can optimize the expected translation quality per unit of user effort by iteratively selecting to supervise the sentence of maximum value of information. Since the term $\mathrm{G}_{\mathcal{L}}$ is independent of the sentence $\mathbf{f}$, the selection for maximizing the value of information can be expressed as the following maximization:

$$\mathbf{f}^* = \arg\max_{\mathbf{f} \in \mathcal{B}} \frac{\mathrm{G}_{\mathcal{L}'}}{\mathrm{E}(\mathbf{f})} \tag{5.2.5}$$

According to this criterion, we can obtain the set $\mathcal{S}$ of sentences to be supervised by repeatedly selecting one more sentence according to the above equation up to the predefined effort level ρ. However, this selection strategy is not adequate for a practical deployment due to it high computational complexity. The exact implementation of the maximization in Equation (5.2.5) requires to loop over all the source sentences in $\mathcal{B}$, and compute for each of them the total expected quality of the automatic translations for the sentences in $\mathcal{B}$, Equation (5.2.2). The computational complexity of each query iteration in such an algorithm is in $O(|\mathcal{B}|^2 \cdot |\mathcal{E}|)$, where $|\mathcal{E}|$ is the number of possible

target language sentences. If we additionally consider the cost of updating the SMT model after each sentence selection and the strict time constraints inherent to the IMT scenario, we conclude that the direct implementation of Equation (5.2.5) as sampling strategy is impractical for IMT.

Alternatively, we propose different sampling strategies defined in terms of a particular utility function $\Phi(\mathbf{x})$ that can be computed very fast. The proposed utility functions cannot typically be derived from the formulation of the VOI framework, but they are designed to exploit the insights provided by the framework about which are the factors that must be taken into account to optimally select the sentences. First, we present a baseline random sampling. Then, we describe two classical active learning approaches, *uncertainty* and *information density*, that assume all automatic translations to have the same supervision cost. Finally, we propose a sampling strategy that explicitly take into account user effort to score the sentences.

Random Ranking (R)

Random ranking, where a random score in the range $[0, 1]$ is assigned to each sentence, is the baseline ranking function used in the experimentation. Although simple, random ranking performs surprisingly well in practice. Its success stems from the fact that it always selects sentences according to the underlying distribution. Using a typical active learning heuristic, as training proceeds and sentences are sampled, the training set quickly diverges from the real data distribution. This difficulty known as *sampling bias* [Dasgupta and Hsu, 2008] is the fundamental characteristic that separates active learning from other learning methods. However, since by definition random ranking selects sentences according to the underlying distribution, it does not suffer form sampling bias. This fact makes random ranking a very strong baseline to compare with.

Uncertainty Ranking (U)

One of the most common active learning sampling strategies is uncertainty sampling [Lewis and Gale, 1994]. This strategy select those samples that the system cannot reliably label. The intuition is clear: much can be learned from the correct output if the model is uncertain of how to label the sample. Formally, a typical uncertainty sampling strategy scores each sample $\mathbf{f}$ with one minus the probability of its most probable prediction $\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \mathrm{P}(\mathbf{e} \mid \mathbf{f})$:

$$\Phi(\mathbf{f}) = 1 - \mathrm{P}(\hat{\mathbf{e}} \mid \mathbf{f}) \tag{5.2.6}$$

Note that the previous equation is an approximation to the value of information in Equation (5.2.4) where the following three assumptions have been taken. First, translation quality is measured by sentence error rate, i.e. $Q(\mathbf{e}, \mathbf{e}')$ is a 0-1 function whose value is equal to one if $\mathbf{e}$ is equal to $\mathbf{e}'$ and zero otherwise. Second, the improvement in translation quality is computed individually for each sentence not taking into account the effect of adding the potential new sentence pair to the model, i.e. $P_{\mathcal{L}}(\mathbf{e}' \mid \mathbf{f}')$ is assumed to be equal to $P_{\mathcal{L}'}(\mathbf{e}' \mid \mathbf{f}')$ for all source sentences $\mathbf{f}' \in \mathcal{B} \setminus \{\mathbf{f}\}$ and all possible translations $\mathbf{e}'$. Third, supervision effort is considered constant for all sentences.

Uncertainty sampling has been applied to a number of different tasks, however, due to the peculiarities of state-of-the-art SMT models, this approach has to be re-considered in our case. Translation models usually do not generate "true" probability distributions but simple scores. Since the normalization term does not influence the decision on the highest-probability translation, it is usually ignored in the model formulation, see Equation (1.3.12). Thus, the scores of two different translations are not directly comparable and the conventional uncertainty technique provides poor performance [Haffari et al., 2009]. Instead, under the assumption that the "certainty" of a model in a particular translation is correlated with the quality of that translation, we measure the uncertainty of a translation using an estimation of its quality. As done for the sentence-level active interaction in Chapter 4 (Section 4.2.2), we estimate the quality of a translation from the quality scores of their individual words.

Given a target language sentence $\mathbf{e} = \mathrm{e}_1 \dots \mathrm{e}_i \dots \mathrm{e}_{|\mathbf{e}|}$ suggested as translation of the source sentence $\mathbf{f} = \mathrm{f}_1 \dots \mathrm{f}_j \dots \mathrm{f}_{|\mathbf{f}|}$, the estimated quality of a target language word e_i is computed as:

$$\Upsilon(\mathbf{f}, \mathrm{e}_i) = \max_{0 \leq j \leq |\mathbf{f}|} P(\mathrm{e}_i \mid \mathrm{f}_j) \tag{5.2.7}$$

where f_0 is the *empty* or *null* word, introduced to capture a target word that corresponds to no actual source word, and $P(\mathrm{e}_i \mid \mathrm{f}_j)$ is the word-to-word lexicon, namely the probability of target word e_i of being the translation of source word f_j. A detailed description of this word-level quality estimator has been provided in Chapter 3 Section 3.5.3.

Then, we choose to compute the quality-based uncertainty score as one minus the ratio of words in the most probable translation $\hat{\mathbf{e}} = \mathrm{e}_1 \dots \mathrm{e}_i \dots \mathrm{e}_{|\hat{\mathbf{e}}|}$ classified as incorrect according to a word-confidence threshold τ_w, see Section 4.2.2:

$$\Phi_{\mathrm{U}}(\mathbf{f}) = 1 - \frac{|\{\mathrm{e}_i \mid \Upsilon(\mathbf{f}, \mathrm{e}_i) > \tau_w\}|}{|\hat{\mathbf{e}}|} \tag{5.2.8}$$

In the experimentation, the value of threshold τ_w was optimized to minimize classification error in a separate development set. Additionally, we use the incremental version of the EM algorithm [Neal and Hinton, 1999] to update the word-to-word probability model $\mathrm{P}(\mathrm{e}_i \mid \mathrm{f}_j)$ with the new sentence pairs available. We thus maintain an updated version of the probability distribution over translations so that the user is not repeatedly asked to supervise translations that provide similar information.

Information Density Ranking (ID)

Uncertainty sentence selection bases its decisions on individual instances which makes the technique prone to sample outliers. The least certain sentences may not be representative of other sentences in the distribution, in this case, knowing its label is unlikely to improve the future accuracy of the model [Roy and McCallum, 2001]. We can overcome this problem by modeling the input distribution explicitly when scoring a sentence.

The information density framework [Settles and Craven, 2008] is a general density-weighting technique. The main idea is that informative instances should not only be those which are uncertain, but also those which are representative of the underlying distribution (i.e., inhabit dense regions of the input space). To address this, we compute the information density score:

$$\Phi_{\mathrm{ID}}(\mathbf{f}) = \Phi_{\mathrm{U}}(\mathbf{f}) \cdot \left(\frac{1}{|\mathcal{B}|} \sum_{b=1}^{|\mathcal{B}|} \mathrm{S}(\mathbf{f}, \mathbf{f}_b) \right)^{\gamma} \tag{5.2.9}$$

where the uncertainty of a given sentence $\mathbf{f}$ is weighted by its average similarity $\mathrm{S}(\mathbf{f}, \cdot)$ to the rest of sentences in the distribution, subject to a parameter γ that controls the relative importance of the similarity term. Since the distribution is unknown, we use the block of sentences $\mathcal{B} = \{\mathbf{f}_1, \ldots, \mathbf{f}_b, \ldots, \mathbf{f}_{|\mathcal{B}|}\}$ to approximate it. Additionally, we use the uncertainty ranking $\Phi_{\mathrm{U}}(\mathbf{f})$ function defined in the previous section to measure the "base" value of a sentence, but we could use any other sentence-level strategies proposed in the literature [Settles and Craven, 2008; Haffari et al., 2009]. As uncertainty sampling, information density simply assumes that all translations have the same supervision cost.

As similarity measure, we use a score that is closely related to the widespread BLEU score [Papineni et al., 2002] presented in Chapter 1 Section 1.6.1. Specifically, we compute the similarity between two sentences as the geometric mean

of the precision of n-grams up to size four between them:

$$\mathrm{S}(\mathbf{f}, \mathbf{f}_b) = \left(\prod_{n=1}^{4} \frac{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{f})} \min(\#_{\mathbf{w}}(\mathbf{f}), \#_{\mathbf{w}}(\mathbf{f}_b))}{\sum_{\mathbf{w} \in \mathcal{W}_n(\mathbf{f})} \#_{\mathbf{w}}(\mathbf{f})} \right)^{\frac{1}{4}} \tag{5.2.10}$$

where $\mathcal{W}_n(\mathbf{f})$ is the set of n-grams of size n in $\mathbf{f}$, and $\#_{\mathbf{w}}(\mathbf{f})$ represents the count of n-gram $\mathbf{w}$ in $\mathbf{f}$.

One potential drawback of information density is that the number of similarity calculations grows quadratically with the number of instances in $\mathcal{B}$. However, the similarities only need to be computed once for a given $\mathcal{B}$ and are independent of the base measure. Thus, we can pre-compute and cache the similarity scores for efficient lookup during the active learning process.

Coverage Augmentation Ranking

Sparse data problems are ubiquitous in natural language processing [Zipf, 1935]. In a machine learning scenario, this means that some rare events will be missing completely from a training set, even when it is very large. Missing events result in a loss of coverage, a situation where the structure of the model is not rich enough to cover all types of input. For out-of-coverage words, an SMT model may not be able to predict any translation at all or only output a generic translation; words (or sequences thereof) that do not appear in the training set cannot be adequately translated [Turchi et al., 2009; Haddow and Koehn, 2012].

According to these considerations, we should explicitly measure the amount of unseen events to improve translation quality in SMT. We do that by measuring the coverage augmentation $\Delta_{\mathrm{cov}}(\mathbf{f}, \mathcal{L})$ due to the incorporation of each sentence $\mathbf{f}$ to the current set of parallel sentences $\mathcal{L}$ used to estimate the SMT model:

$$\Delta_{\mathrm{cov}}(\mathbf{f}, \mathcal{L}) = \sum_{n=1}^{4} \sum_{\mathbf{w} \in (\mathcal{W}_n(\mathbf{f}) - \mathcal{W}_n(\mathcal{L}))} \sum_{b=1}^{|\mathcal{B}|} \#_{\mathbf{w}}(\mathbf{f}_b) \tag{5.2.11}$$

The coverage augmentation for each sentence $\mathbf{f}$ is given by the count of n-grams missing in the training set $\mathcal{L}$ that appear in the rest of sentences in the block. In other words, we measure how many missing n-grams in $\mathcal{B}$ will be covered if $\mathbf{f}$ is added to $\mathcal{L}$. Again, we consider $n = 4$ as the maximum n-gram length. Lastly, we break the potential ties by selecting the longest sentence.

Note that coverage augmentation estimates the value of each sentence in terms of the number of new events it contains weighting the count of each n-gram according to the number of times it appears in the rest of sentences. Therefore, $\Delta_{\text{cov}}(\mathbf{f}, \mathcal{L})$ can be seen as an information density ranking method that jointly estimates value and information density.

This coverage augmentation score is biased towards long sentences since longer sentences can contain more unseen n-grams. This is one of the reasons for the successful application of this idea in conventional AL scenarios [Haffari et al., 2009] and bilingual sentence selection tasks [Gascó et al., 2012]. However, longer sentences also imply a higher supervision effort from the user [Koponen, 2012] which may penalize performance. We address this trade-off by normalizing the coverage augmentation score by an estimation of the user-effort $\text{E}(\mathbf{f}, \mathcal{L})$ required to supervise the translation. Since out-of-coverage words cannot be adequately translated and their translations will be corrected by the user, we assume user effort to be proportional to the number of new n-grams in the source sentence:

$$\text{E}(\mathbf{f}, \mathcal{L}) \propto \sum_{n=1}^{4} \sum_{\mathbf{w} \in (\mathcal{W}_n(\mathbf{f}) - \mathcal{W}_n(\mathcal{L}))} \#_{\mathbf{w}}(\mathbf{f}) \tag{5.2.12}$$

Finally, the coverage augmentation score measures the potential SMT model improvement per unit of user effort[b]:

$$\Phi_{\text{CA}}(\mathbf{f}, \mathcal{L}) = \frac{\Delta_{\text{cov}}(\mathbf{f}, \mathcal{L})}{\text{E}(\mathbf{f}, \mathcal{L})} \tag{5.2.13}$$

In contrast to uncertainty and information density, coverage augmentation does take explicitly into account the supervision effort needed by each individual sentence. An additional difference is that the coverage augmentation score depends solely on the source sentence. That is, it is independent of the particular SMT model. Regarding the value of information in Equation (5.2.4), coverage augmentation can be seen as a practical approximation where where both the improvement in expected translation quality and the supervision effort are measured in terms of the number of the previously-unseen n-grams that appear in the sentences in $\mathcal{B}$.

To avoid selecting several sentences with the same missing n-grams, we update the set of n-grams seen in training each time a new sentence is selected. First, sentences in $\mathcal{B}$ are scored using Equation (5.2.13). Then, the highest-scoring sentence is selected and removed from $\mathcal{B}$. The set of training n-grams

[b] We ignore the effort proportionality constant since we assume it equal for all sentences.

is updated with the n-grams present in the selected sentence and, hence, the scores of the rest of the sentences in the block are also updated. This process is repeated until we select the desired ratio ρ of sentences from $\mathcal{B}$.

5.2.3 On-line Training for SMT

After the translation supervision process, we have a new sentence pair $(\mathbf{f}, \mathbf{e})$ at our disposal. We now briefly describe the incremental SMT model used in the experimentation, and the on-line learning techniques implemented to update the model with new sentence pairs in constant time.

We implement the on-line learning techniques proposed in [Ortiz-Martínez et al., 2010]. In that work, a state-of-the-art log-linear SMT model [Och and Ney, 2002] was presented. This model is composed of a set of incremental feature functions governing different aspects of the translation process, see Equation (1.3.11), including a language model, a model of source sentences length, direct $\mathrm{P}(\mathbf{e} \mid \mathbf{f})$ and inverse $\mathrm{P}(\mathbf{f} \mid \mathbf{e})$ phrase-based translation models [Koehn et al., 2003], models of the length of the source and target language phrases, and a reordering model. Appendix D provides a detailed description of these incremental features.

Together with this log-linear SMT model, Ortiz-Martínez et al. [2010] present on-line learning techniques that, given a training pair, update the incremental features in constant time. In contrast to conventional batch learning techniques, the computational complexity of adding a new training pair does not depend on the number of training samples that have been previously seen. To do that, a set of sufficient statistics is maintained for each feature function. If the estimation of the feature function does not require the use of the EM algorithm [Dempster et al., 1977] (e.g. language models), then it is generally easy to incrementally update the feature given a new training sample. By contrast, if it is required (e.g. to estimate phrase-based SMT models), the estimation procedure has to be modified because the conventional EM algorithm is designed for its use in batch learning scenarios. For such feature functions, the incremental version of the EM algorithm [Neal and Hinton, 1999] is applied. A detailed description of the update algorithm for each feature function in the log-linear SMT model can be found in [Ortiz-Martínez, 2011].

5.3 Experiments

Now, we describe the series of experiments carried out to assess the soundness of the proposed active learning framework. The idea is to simulate a real-

corpus	use	sentences	tokens (Spa/Eng)	vocabulary (Spa/Eng)	out-of-coverage tokens (Spa/Eng)
Europarl	training	731k	15.7M/15.2M	103k/64k	–/–
	tuning	2k	60k/58k	7k/6k	208/127
News Commentary	test	51k	1.5M/1.2M	48k/35k	13k/ 11k

Table 5.1: Main figures of the Spanish– English corpora used, k and M stand for thousands and millions of elements respectively.

world scenario where a translation agency is hired to translate a huge amount of text, then we study the productivity, i.e. the ratio between translation quality and user effort, obtained by different setups of the proposed active learning framework. The experimentation was divided into two parts. First, Section 5.3.3 describes a conventional active learning experimentation where we studied the learning curves of the SMT model as a function of the number of training sentence pairs. Then, Section 5.3.4 focuses on user effort and studies the productivity, i.e. the ratio between translation quality and user effort, obtained by each setup of the proposed framework.

5.3.1 Methodology and Data

The experimentation performed comprised the translation of a test corpus using different setups of the proposed active learning approach. Each setup was defined by the ranking function used. All experiments started with a "base" log-linear SMT model trained on the Europarl [Koehn and Monz, 2006] corpus. We used the training part of the Europarl corpus to train the feature functions of the model, and the development part to estimate the values of the log-linear weights (see Equation (1.3.11)) by means of minimum error-rate training [Och, 2003] optimizing BLEU [Papineni et al., 2002].

Once the "base" SMT model had been trained, we translated the News Commentary corpus [Callison-Burch et al., 2007] following Algorithm 5.1 using the different sampling strategies presented in Section 5.2.2. We used blocks of size $|B| = 1000$, and for information density, we arbitrarily set $\gamma = 1$ (i.e., uncertainty and density terms had equal weight). The main figures of the training, tuning, and test corpora are shown in Table 5.1.

The reasons to choose the News Commentary corpus to be translated with the proposed active learning algorithm for IMT are threefold: its size is large

enough to test the proposed techniques in the long term, it contains sentences from a different domain than the sentences in the training and tuning corpora, and lastly, it consists in editorials from different domains which allow us to test the robustness and adaptability of our system against domain-changing data streams. Thus, by translating the News Commentary corpus we were simulating a realistic scenario where translation agencies must be ready to fulfill eclectic requests for translation.

5.3.2 Evaluation Measures

Our goal is to generate translations of maximum quality while minimizing the human effort required to generate them. The evaluation thus was both in terms of translation quality and human supervision effort. We present translation quality results as measured by BLEU [Papineni et al., 2002] (see Section 1.6.1) and user effort results as measured by KSMR [Barrachina et al., 2009] (see Section 1.6.2).

We also present a study on the statistical significance of the pairwise performance differences observed. In this case, to avoid the multiple comparisons problem [Miller, 1966], we measure the statistical significance of the observed differences using Tukey's honest significant difference [Hsu, 1996] (HSD) tests. Specifically, we implement a modified version of the randomized paired-samples t-test described in Section 1.6.3 so that the obtained p-values are adjusted for multiple comparisons by Tukey's HSD [Carterette, 2012].

5.3.3 Conventional Active Learning Results

We first studied the different ranking functions in a typical active learning experimentation where prediction accuracy is measured as a function of the number of examples used to update the learning model. The performance of the SMT model was measured as the translation quality (BLEU) of the initial automatic translations generated during the interactive supervision process (line 6 in Algorithm 5.1).

Figure 5.1 displays the learning rates observed for the ranking functions described in Section 5.2.2: random (R), uncertainty (U), information density (ID) and coverage augmentation (CA). For coverage augmentation, we also show both the learning rate obtained using directly the Δ_{cov} scoring function in Equation (5.2.11). In addition to these learning rates, we report significance level of the difference for some pairwise comparisons between results of the same ρ value. We give p-values on a logarithmic scale and mark two standard levels of significance, 0.01 and 0.05, for reference.

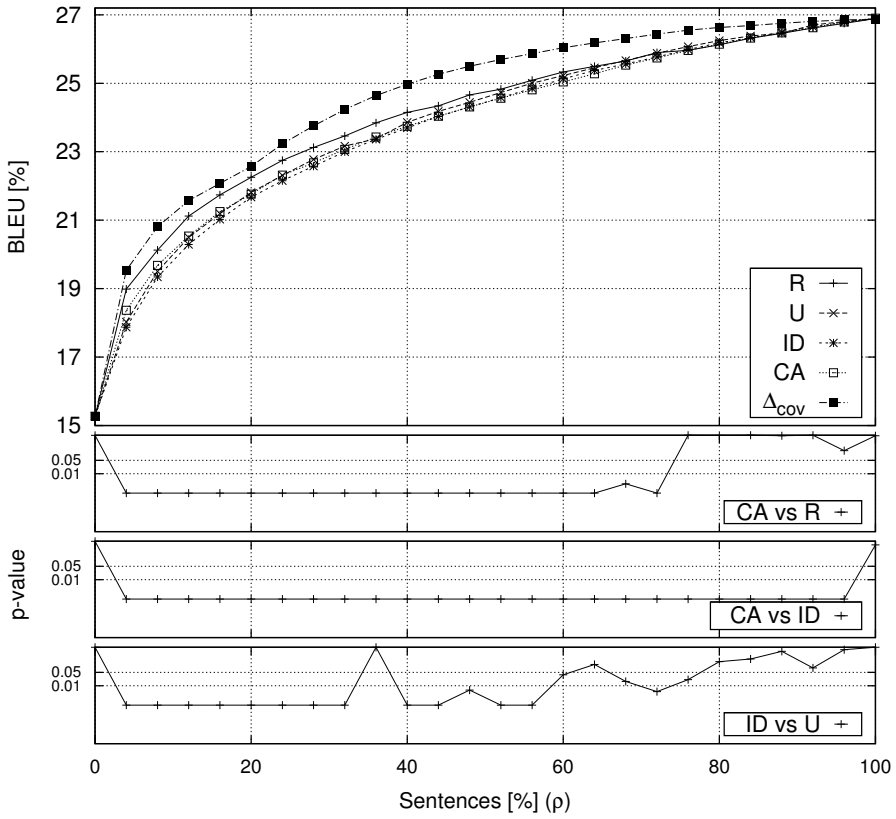


Figure 5.1: Translation quality (BLEU) of the translations generated by the SMT model updated with the sentence pairs sampled by different ranking functions. Performance is displayed as a function of the percentage ρ of the corpus used to update the SMT model. We also present Tukey's HSD p-values for some pairwise comparisons between the results observed for different rank functions.

Figure 5.1 shows that random ranking is a quite strong baseline. It outperformed coverage augmentation, information density and uncertainty sampling, although the observed differences were scarce. Up to a 50% of supervised sentences the performance difference was statistically significant; after that, results for the four ranking functions were very similar and almost no statistical difference can be observed (second panel). In contrast, results showed that Δ_{cov} consistently outperformed all other ranking functions. Additionally, the observed difference is statistically significant as shown in the third panel of the figure. Lastly, Uncertainty ranking and information density ranking obtained

virtually the same results; however the slightly better results of uncertainty ranking were statistically significant (fourth panel).

These results confirm the intuition followed when designing the coverage augmentation scoring function. Measuring the number of unreliable modeled events, n-grams in our case, is a good estimator of the potential improvement of the SMT model. We hypothesize that this is due to the intrinsic sparse nature of natural language, and particularly by the eclectic domains, e.g. economic, science, or politics, of the sentences in the test set. These results are also coherent with previous works on active learning for SMT [Haffari et al., 2009] and confirms the good results that the application of this idea has obtained in other natural language processing tasks, for example [Gascó et al., 2012].

Finally, it is worth notice the quite important difference that was observed between Δ_{cov} and coverage augmentation. These strategies only differ in the effort estimation: Δ_{cov} assumes all sentences require the same human effort to be supervised while coverage augmentation explicitly estimates the supervision effort of each sentence. Thus, the observed performance differences are bound to be due to the effect of the particular effort estimation. Δ_{cov} was able to select any sentence regardless the effort required to supervise it, which explains the good performance of this strategy. However, those sentences that contain most new n-grams are also those that require more supervision effort from the user. Taking this into account, coverage augmentation was bound to select those sentences that provide a better value / effort ratio.

5.3.4 Cost-Sensitive Active Learning Results

The experiments in the previous section assumed that all automatic translations require the same supervision effort from the user. However, it is clear that different sentences require different supervision costs. Thus, next we focus on measuring the human effort required to supervise the translation of the sentences sampled by the different ranking functions. Figure 5.2 shows the KSMR scores obtained by each ranking function as a function of the percentage of sentences ρ for which the system asked for user supervision. Additionally, we display the significance of the performance differences observed for some pairwise ranking function comparisons.

We can observe that the sentences sampled by coverage augmentation consistently required less human effort than any other ranking function. Additionally, these differences were significant as showed in the second panel of the figure. In contrast, the sentences sampled by Δ_{cov} required much more human effort to supervise them. For instance, when supervising a 40% of the sentences the ones selected by Δ_{cov} required almost the double of human effort

than the ones selected by coverage augmentation. This result came to confirm the intuition stated above to explain the results obtained by Δ_{cov} and coverage augmentation in the conventional active learning experiments in Figure 5.1. The huge difference in human effort between these two sampling strategies assess the effectiveness of the proposed effort normalization implemented in coverage augmentation, see Equation (5.2.13). Clearly, these differences in user effort were statistically significant (third panel). Lastly, uncertainty and information density required a lower amount of effort than random, and similarly to the results in Figure 5.1, the observed differences between them were scarce but statistically significant (fourth panel). In this case, sentences selected by information density required a statistically lower amount of effort to be supervised.

A particularly interesting phenomena occurs when all sentences are supervised $\rho = 100\%$. We can observe that uncertainty, information density, coverage augmentation and even Δ_{com} required a slightly lower amount of effort than random sampling. This fact contradicted the common intuition by which if all sentences are supervised the human effort should be equal regardless of the chosen ranking function. This counter-intuitive result is due to the different order (depending on the ranking function) in which the translations in each block are supervised. In other words, despite supervising all sentences, we can reduce the human effort by supervising in first place those sentences considered more valuable.

Up to now, we have confirmed that the ranking functions that obtain better translation quality in conventional active learning experiments tend to require more effort from the user to supervise their sampled sentences. The comparison between Δ_{cov} and coverage augmentation is very illustrative, by modeling supervision effort (instead of assuming it constant) we were able to obtain similar improvements in the performance of the SMT model (about one BLEU points of difference) with only half the user effort. Back to the point of view of a translation agency with limited resources, the key objective has always been to obtain the best translation productivity. Therefore, we want to achieve the best compromise possible between the quality of the final translations generated and the human effort required to obtain them. In other words, given a level of translation quality, we want to minimize the required supervision effort; or symmetrically, given an effort level, we aim at maximizing translation quality.

To address this challenge, we studied the trade-off between user effort and translation quality that can be achieved with each ranking function. In contrast with the experimentation in Figure 5.1 where we study the evolution

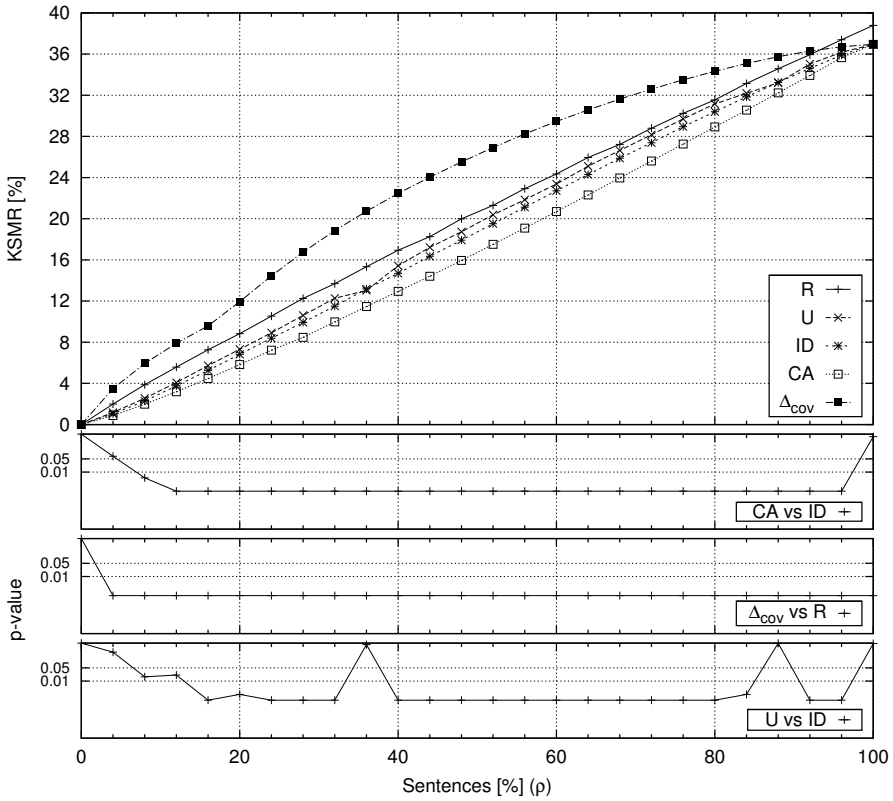


Figure 5.2: User effort (KSMR) required to supervise the translations sampled by different ranking functions. User effort is displayed as a function of the percentage ρ of the corpus used to update the SMT model. We also present Tukey's HSD p-values for some pairwise comparisons between the results observed for different rank functions.

of the translation quality of the automatic translations generated by the SMT model, we now are interested in the performance of the complete IMT system with active learning. We did that by measuring the translation quality of the translations outputted by Algorithm 5.1 (lines 15 and 17) as a function of the required supervision effort. Note that these final translations are a mixture of automatic and user-supervised translations. The ratio between them is fixed by ρ which permits to adjust the system's behavior between that of a fully automatic SMT system if none translation is supervised ($\rho = 0\%$), or that of a conventional IMT system where all translations are supervised by the user ($\rho = 100\%$).

Given the similar performance reported by uncertainty and information

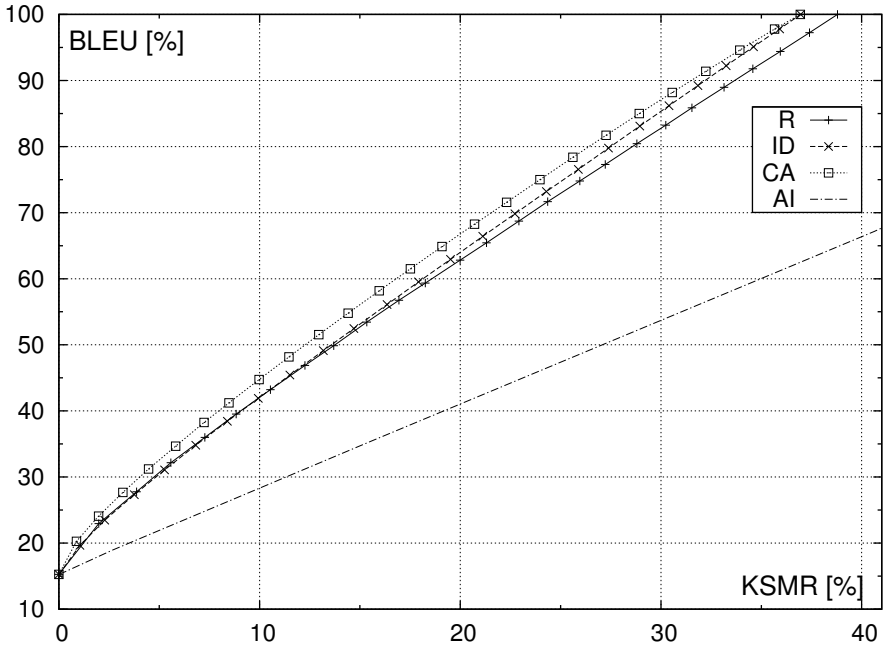


Figure 5.3: Quality (BLEU) of the translations generated by the proposed IMT system with active learning as a function of user effort (KSMR) required to generate them. We study different ranking functions, and provide comparative results of a sentence-level active-interaction (AI) system (see Chapter 4) that does not implement SMT model updating.

density in the previous experiments and the huge different in human effort between Δ_{cov} and coverage augmentation, Figure 5.3 compares the performance of random (R), information density (ID), and coverage augmentation (CA). Additionally, we present results of a sentence-level active interaction (AI) system such as the one studied in Chapter 4. Since this system does not update the SMT model with the translations supervised by the user, we used its results as a baseline to test the influence that the update of the SMT model has on translation productivity

The first that we can observe in Figure 5.3 is the huge leap in productivity that was obtained when the SMT model was updated with user feedback. The continuous model updating allowed us to obtain translations of almost twice the quality with the same amount of effort in comparison to active interaction. Regarding the different ranking functions, random ranking obtained the lowest productivity ratio with larger differences with respect to the rest of

ranking functions for high levels of effort. In contrast, the proposed coverage augmentation ranking consistently obtained the best trade-offs between final translation quality and required human effort. For instance, given an effort level of 20 KSMR points coverage augmentation generated better translations (over 5 BLEU points) that random sampling. Lastly, information density obtained a performance between random ranking and coverage augmentation. For low levels of effort its results were similar to those by random ranking, however, as more human effort was invested its performance slowly got close to that of coverage augmentation.

5.4 Summary

We have presented an active learning framework designed to boost the translation productivity of IMT systems. The two cornerstones of our approach are an active interaction protocol to selectively supervise the translations and a dynamic SMT model that is continually updated with user-supervised translations. Regarding the active interaction protocol, we have propose a selective supervision protocol where the user only supervises the automatic translation of a subset of the source sentences. The sentences to be supervised are selected so that they maximize a utility function. We propose diverse functions to measure the utility of the sentences. The percentage of sentences to be supervised is defined by a tunable parameter which allows to adapt the system to met task requirements in terms of translation quality, or resources availability. Then, whenever a new user-supervised translation pair is available, we use it to incrementally update a dynamic SMT model. A set of on-line learning techniques habe been implemented to update the model in constant time. Despite being studied and implemented for an IMT system, this is a general active learning framework that can be applied to other CAT approaches.

We have evaluated the proposed active learning framework in a scenario where we intended to simulate the translation requirements that a real-world translation agency may receive. We divided the experimentation into two parts: experiments in a conventional active learning scenario, and experiments in a cost-sensitive scenario. The results of the experiments showed that those utility functions that improved most the quality of the underlying SMT model were usually the same that required the most user effort. However, by modeling the user supervision effort (instead of considering it constant as conventional active learning methods do), one of the studied techniques, coverage augmentation, was able to achieve the best trade-offs between the quality of the generated translations and the human effort required to obtain them. In

comparison to an active interaction protocol with no SMT model updating, coverage augmentation obtained almost twice the translation quality for the same amount of effort. This result shows the crucial importance of model retraining to boost the productivity of IMT systems. Finally, coverage augmentation also outperformed all other utility functions including the usually strong random baseline. For instance, for the same effort level, it was able to obtain better translations (more than five BLEU points) in comparison to random sampling.

Chapter

6 Conclusions

This final chapter presents a summary of the scientific contributions achieved by this thesis. For each of the research lines exposed in the previous chapters, we describe the work carried out and the results obtained. We also provide a list with the publications derived from the work carried out in the different research lines. Finally, we identify research directions that are worth of being explored in future developments.

Chapter Outline

6.1 Scientific Contributions

We have explored three different research directions to improve the broader and more efficient deployment of current MT technology. Following sections describe the scientific contributions accomplished on each direction.

6.1.1 Combination of Machine Translation Systems

In this first line of research, we have focused on the improvement of fully-automatic MT technology. To do that, we chose to investigate on methods to automatically combine the outputs of multiple MT systems into a consensus translation of higher quality.

As a result of the work carried out, we have developed a new system combination method for MT named minimum Bayes' risk system combination (MBRSC). MBRSC is able to detect the high-quality subsequences in the provided translations and combine them into a consensus translation of maximum expected BLEU score. We have formalized MBRSC as a weighted ensemble that combines the probability distributions over translations of the individual MT systems. Then, we have derived the optimum minimum Bayes' risk decision function for such ensemble model. As loss function, we have chosen the most widespread translation quality measure: the BLEU score. Finally, we have also described a minimum error rate training method to learn the optimal values of the weights in the ensemble.

The direct implementation of this optimal decision function has shown to be unfeasible due to its high complexity. Thus, we have proposed several approaches to efficiently obtain the optimal consensus translation. To do that, we have split the search problem into two closely related sub-problems: the computation of the risk for a given translation candidate, and the actual search for the optimal consensus translation.

Regarding the computation of the risk, we have studied two different alternatives to the exact risk computation. On the one hand, we have implemented this formulation to compute the exact value of the risk for a linear approximation to the BLEU score. On the other hand, we have implemented an approximate computation of the risk, based on expected n-gram counts, for the exact BLEU score.

We have also proposed different alternatives to the exhaustive search for the consensus translation among the (potentially) infinite number of target language sentences. We have started by describing a greedy gradient ascent search algorithm that takes as input a candidate translation that is iteratively improved by the application of different edit operations. Then, we have

described a search algorithm based on dynamic programming. For the risk based on the linear approximation to BLEU score, this dynamic programming search can be implemented exactly. In contrast, the higher complexity of the BLEU risk over expected counts made its dynamic programming formalization impractical. Thus, we have finally implemented it by a beam search algorithm with pruning. Lastly, we have also provided a complexity analysis for the particular formulation of each search algorithm depending on the risk computation method used.

We have also present the results of a thorough empirical study of MBRSC. First, we conducted a series of comparative experiments to determine the best combination of risk computation method and search algorithm. Results showed that BLEU over expected counts was the best risk computation method obtaining virtually the same results as the exact BLEU risk, and consistently outperforming linear BLEU for all search algorithms. Then, we conducted further experiments to determine the best search algorithm. The results of this experimentation showed that beam search outperformed the other algorithms being, for instance, more efficient than gradient ascent search. Finally, we compared the optimal MBRSC setup (risk over expected counts and beam search) to different state-of-the-art MT system combination methods. Results showed that the performance of MBRSC is comparable to that of the other methods. Moreover, since MBRSC generates the consensus translation directly from the provided translations, its performance is not limited by the availability of additional data which makes MBRSC a particularly well suited method to be applied to languages with scarce resources.

6.1.2 Machine Translation Quality Estimation

The goal of this second research direction has been to improve the utility of automatic translations for the end-user. To do that, we have investigated on methods to automatically estimate at run-time the quality of such translations.

We have presented a two-step training methodology for regression models whose goal is to efficiently manage the noisy and collinear features usually computed to predict the quality of natural language sentences. Our proposal divides training into two steps: a dimensionality reduction step, and the actual estimation of the regression model from the reduced feature set. We have proposed two novel dimensionality reduction methods based on the PLSR model, and studied several other DR methods previously used in the literature. The DR methods under consideration can be classified by their theoretical background: statistical multivariate analysis or heuristic methods, or by how they perform the reduction: feature selection or feature extraction methods.

Additionally, we have also studied different regression models and how DR influences their prediction accuracy.

We have performed a thorough empirical evaluation to assess the soundness of the proposed two-step training methodology. Initially, we have evaluated each DR method by the prediction accuracy of the regression systems trained on the corresponding reduced feature set. The results of this experimentation showed that feature extraction methods usually outperformed feature selection methods, and that the performance of the different DR methods was to a great extent independent of the chosen regression model. Among the different DR methods, one of the proposed methods, PLS-P, obtained the best performance both in terms of prediction accuracy and feature reduction ratios.

We then performed a second series of experiments to exhaustively test PLS-P on different feature sets. These experiments allowed us to evaluate the proposed methodology in a wide range of conditions. These results showed that PLS-P was indeed able to strip out the noise present in the original feature sets, and at the same time, these reduced feature sets outperformed the prediction accuracy of the whole original feature sets. Additionally, the reduced feature sets extracted by PLS-P also improved the prediction accuracy of those extracted with the widespread PCA method.

Finally, one of the advantages of the proposed training methodology was that it reduced the operating time of the QE system. Hence, we could take advantage of this efficiency to predict translation quality from hundreds of features. Results showed that PLS-P was able to efficiently manage more than a thousand features to largely improve prediction accuracy. Alternatively, this time-efficiency makes this approach well-suited to be deployed in scenarios with strict temporal restrictions such as IMT.

6.1.3 Active Protocols for Interactive Machine Translation

The goal of this third research direction has been the development of methods to improve the usability, and thus the productivity, of computer-assisted translation technology, specifically to improve the productivity of interactive machine translation (IMT) systems. The key of our approach has been the concept of active protocol. In contrast to passive protocols where the user is assumed to systematically supervise all translations, in an active protocol the system proactively informs the user about which translation elements should undergo user supervision. First, we have studied an active interaction protocol where the system informs the user about the reliability of the translations suggested by the system. Then, we have proposed an active learning framework for IMT where the system additionally learns from user feedback.

Active Interaction

We have proposed an active interaction protocol where the IMT system informs the user about the reliability of the translations suggested by the system. The key idea was to help the user by focusing his attention to those translation elements more likely to be incorrect, hence easing the We have studied such an active interaction protocol both at the word and sentence level. Regarding word-level active interaction, we implement it using a confidence measure based on a word-to-word lexicon. This is a simple confidence measure that fulfills the strict time constraints inherent to the real-time IMT scenario, and additionally, it has been reported to provide very good classification accuracy. This word-level confidence measure is also the basis of the active interaction protocol at the sentence level. In this case, the confidence of a suggested translation was computed as a combination of its word level confidence scores.

Regarding the empirical evaluation, a thorough experimentation involving human users would have been very costly. Therefore, we chose to use a simulated user to extensively evaluate the proposed active interaction protocol. In a first in-laboratory experiment, we studied the accuracy of the proposed word confidence measure in predicting which words would be corrected by an actual human user. Results showed that the confidence information provided a more accurate prediction than considering all words erroneous (as a conventional IMT system does) or considering all words correct (as an SMT system does). In a second in-laboratory experiment, we studied the influence of word- and sentence-level confidence information in the productivity of the IMT system. Results for our simulated user showed that large effort reductions were obtained while still generating high quality translations.

Lastly, we also carried out a small experimentation involving human users. In this experiment, we focused on the proposed word-level active interaction protocol, and more specifically, we studied to which extent actual human users were able to take advantage of the provided reliability information. This qualitative experimentation showed that users considered active interaction as a desirable feature. However, the confidence measure used in the current implementation was perceived as too error-prone which annoys the users and penalized the usability of the system. Nevertheless, users reckoned word-level active interaction as a promising approach if a more reliable reliability information were to be deployed.

Active Learning

We have explored a second active protocol for IMT. In this case, our motivating scenario has been such of a translation agency with limited resources that must fulfill as much requirements for translation as possible. Hence, our research has been focused on the translation productivity of IMT systems. In this context, we measured productivity as a combination of the number and the quality of the translations that can be generated by unit of user effort. We have formalized these ideas as an active learning framework for IMT. In this framework, the IMT system is able to suggest which automatic translations should be supervised by the user, and additionally its underlying SMT model is continually learning from the user supervisions in order to improve future translation suggestions. This active learning framework can be seen as an extension of the active interaction protocol presented beforehand. Additionally, it is a general framework that, in addition to IMT, can be applied to other CAT approaches.

The two cornerstones of our active learning framework are a selective supervision protocol and a continual SMT model updating with user-supervised translations. Regarding selective supervision, we have provided a formal derivation for the value of asking the user to supervise a particular automatic translation. Unfortunately, this approximation had a high computation cost which made impractical its implementation at large scales. Then, we have proposed several ranking functions that aim at approximating the utility of asking the user to supervise a particular translation. We used these user-supervised translations to update the SMT model used by the IMT system. To do that in real-time as required by the IMT scenario, we implemented a state-of-the-art log-linear SMT model and different on-line learning techniques that allowed us to update the model in constant time.

The empirical evaluation of the proposed active learning framework for IMT has been twofold. On the one hand, we have evaluated the performance of the different ranking functions in a conventional active learning experimentation. These initial results showed that different translations require different supervision cost, and those that improve most the performance of the SMT model are the same that require more supervision effort. Since our goal is to boost translation productivity of the IMT system, we carried out a second experiment where we studied the ratio between the quality of the final translations and the human effort required to generate them. Results showed that the proposed active learning framework halved the human effort required to obtain translations of a certain quality in comparison to conventional IMT technology.

6.2 Publications

Now, we summarize the list of publications derived from the work carried out in this thesis. Listed publications are grouped by research line and ordered within by year of publication.

6.2.1 Combination of Machine Translation Systems

The development of MBRSC, the system combination method for MT presented in Chapter 2, were described in various international conferences:

- Jesús González-Rubio and Francisco Casacuberta. On the Use of Median String for Multi-Source Translation. *Proceedings of 20th International Conference on Pattern Recognition*, pp. 4328–4331, 2010. Core B

- Jesús González-Rubio, Alfons Juan and Francisco Casacuberta. Minimum Bayes-risk System Combination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1268–1277, 2011. Core A

Additionally, MBRSC has been compared against other system combination methods in various international competitions:

- Jesús González-Rubio, Jesús Andrés-Ferrer, Germán Sanchis-Trilles, Guillem Gascó, Pascual Martínez-Gómez, Martha A. Rocha, Joan A. Sánchez and Francisco Casacuberta. UPV-PRHLT Combination System for WMT 2010. *Proceedings of the 5th Workshop on Statistical Machine Translation (ACL)*, pp. 296-300, 2010.
  Workshop in Core A conference

- Jesús González-Rubio and Francisco Casacuberta. The UPV-PRHLT combination system for WMT 2011. *Proceedings of the 6th Workshop on Statistical Machine Translation (EMNLP)*, pp. 140–144, 2011.
  Workshop in Core A conference

6.2.2 Machine Translation Quality Estimation

The two-step training methodology proposed in Chapter 3 were described in an international journal article:

- Jesús González-Rubio, José R. Navarro-Cerdan and Francisco Casacuberta. Dimensionality reduction methods for machine translation quality estimation. *Machine Translation*, 2013.

The proposed methodology has also been compared to other QE methods in an international competition:

- Jesús González-Rubio, Alberto Sanchis and Francisco Casacuberta. PRHLT Submission to the WMT12 Quality Estimation Task. *Proceedings of the 7th Workshop on Statistical Machine Translation (NAACL)*, pp. 104–108, 2012.
  Workshop in Core A conference

Additionally, although not reported in this thesis, we have also studied different QE methods at the word-level:

- Jesús González-Rubio, José R. Navarro-Cerdan and Francisco Casacuberta. Partial Least Squares for Word Confidence Estimation in Machine Translation. *Proceedings of the 6th Iberian Conference on Pattern Recognition and Image Analysis*, Volume 7887 of Lecture Notes in Computer Science, pp. 500-508, 2013. Core C

6.2.3 Active protocols for IMT

Regarding the active protocols for IMT described in Chapters 4 and 5, the key ideas of both approaches were presented in two international conferences, respectively:

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. On the Use of Confidence Measures within an Interactive-predictive Machine Translation System. *Proceedings of 14th Annual Conference of the European Association for Machine Translation*, 2010. Core B

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. Balancing User Effort and Translation Error in Interactive Machine Translation Via Confidence Measures. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 173–177, 2010. Core A

The active interaction protocol described in Chapter 4 have been integrated in a computer-assisted translation workbench. A description and an evaluation of the workbench have been published in various international conferences:

- Daniel Ortiz-Martínez, Germán Sanchis-Trilles, Francisco Casacuberta, Vicent Alabau, Enrique Vidal, José-Miguel Benedí, Jesús González-Rubio, Alberto Sanchis and Jorge González. The CASMACAT Project: The Next Generation Translator's Workbench. *Proceedings of the iberSPEECH conference*, 2012.

- Vicent Alabau, Jesús González-Rubio, Luis Leiva, Daniel Ortiz-Martínez, Germán Sanchís-Trilles, Francisco Casacuberta, Barto Mesa-Lao, Ragnar Bonk, Michael Carl and Mercedes García-Martínez. User evaluation of advanced interaction features for a computer-assisted translation workbench. *Proceedings of the Machine Translation Summit XIV*, 2013. Core B

- Vicent Alabau, Ragnar Bonk, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes Garcia-Martínez, Philipp Koehn, Luis Leiva, Bartolome Mesa-Lao, Herve Saint-Amand, Chara Tsoukala, German Sanchis, Daniel Ortiz and Jesus Gonzalez-Rubio. Advanced Computer Aided Translation with a Web-Based Workbench. *Proceedings of the workshop on post-editing technology and practice (MT Summit XIV)*, 2013. Workshop in Core B conference

A thorough description of this computer-assisted translation workbench has also been published in an international journal:

- Vicent Alabau, Ragnar Bonk, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Jesús González-Rubio, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Germán Sanchis and Chara Tsoukala. CASMACAT: An Open Source Workbench for Advanced Computer Aided Translation. *The Prague Bulletin of Mathematical Linguistics*, 2013.

Further developments of the active learning framework described in Chapter 5 were published in two international conferences and an international journal.

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. An Active Learning Scenario for Interactive Machine Translation. *Proceedings of the 13th International Conference on Multimodal Interaction*, 197-200, 2011. Core B

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. Active learning for interactive machine translation. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 245–254, 2012. Core A

- Jesús González-Rubio and Francisco Casacuberta. Cost-Sensitive Active Learning for Computer-Assisted Translation. *Pattern Recognition Letters*, 2013. JCR

Additionally, the work on active learning for IMT was also published as part of two book chapters:

- Jorge Civera, Jesús González-Rubio and Daniel Ortiz-Martínez. Interactive Machine Translation. *Multimodal Interactive Pattern Recognition and Applications*. Springer. 2011. Alejandro H. Toselli, Enrique Vidal and Francisco Casacuberta (Editors).

- Daniel Ortiz-Martínez and Ismael García-Varea. Incremental and Adaptive Learning for Interactive Machine Translation. *Multimodal Interactive Pattern Recognition and Applications*, Springer, 2011. Alejandro H. Toselli, Enrique Vidal and Francisco Casacuberta (Editors).

6.2.4 Additional Research Directions

Finally, different research directions were explored in parallel to the ones described in this thesis. The following publications account for them:

- Jesús González-Rubio, Adrià Giménez-Pastor, Jorge González, Antonio L. Lagarda, José R. Navarro-Cerdan, Laura Eliodoro, Víctor J. Fèlix, Piedachu Peris and Francisco Casacuberta. Una evaluación exhaustiva de SisHiTra, un paradigma híbrido en Traducción Automática. *Proceedings of the IV Jornadas en Tecnologías del Habla*, pp. 93–98, 2006.

- Jesús González-Rubio, Jorge González, Adrià Giménez-Pastor, Antonio L. Lagarda, José R. Navarro-Cerdan and Francisco Casacuberta. Translation applications under the SisHiTra framework. *Proceedings of the 3rd Language & Technology Conference*, pp. 453–457, 2007.

- Jesús González-Rubio, Germán Sanchis-Trilles, Alfons Juan and Francisco Casacuberta. A Novel alignment model inspired on IBM Model 1. *Proceedings of the 12th conference of the European Association for Machine Translation*, pp. 47–56, 2008. Core B

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. Optimization of Log-linear Machine Translation Model Parameters Using SVMs. *Proceedings of the 8th workshop on Pattern Recognition in Information Systems*, pp. 48–56, 2008. Core C

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. Minimum Error-Rate Training in Statistical Machine translation using SVMs. *Proceedings of the 4th Iberian Conference on Pattern Recognition*

and Image Analysis, Volume 5524 of Lecture Notes in Computer Science pp. 378–385, 2009. Core C

- Jesús González-Rubio, Jorge Civera, Alfons Juan and Francisco Casacuberta. Saturnalia: A Latin-Catalan Parallel Corpus for Statistical Machine Translation. *Proceedings of the 7th international conference on Language Resources and Evaluation*, pp. 3405–3408, 2010. Core C

- Germán Sanchis-Trilles, Jesús Andrés-Ferrer, Guillem Gascó, Jesús González-Rubio, Pascual Martínez-Gómez, Martha-Alicia Rocha, Joan-Andreu Sánchez and Francisco Casacuberta. UPV-PRHLT English–Spanish system for WMT 2010. *Proceedings of the 5th Workshop on Statistical Machine Translation (ACL)*, pp. 172–176, 2010.
Workshop in Core A conference

- Guillem Gascó, Vicent Alabau, Jesús Andrés-Ferrer, Jesús González-Rubio, Martha-Alicia Rocha, Germán Sanchis-Trilles, Francisco Casacuberta, Jorge González and Joan-Andreu Sánchez. ITI-UPV system description for IWSLT 2010. *Proceedings of the International Workshop on Spoken Language Translation*, 2010.

- Jesús González-Rubio, Daniel Ortiz-Martínez and Francisco Casacuberta. Fast incremental active learning for statistical machine translation. *Avances en Inteligencia Artificial, proceedings of the Conferencia de la Asociación Española para la Inteligencia Artificial*, 2011.

- Germán Sanchis-Trilles, Daniel Ortiz-Martínez, Jesús González-Rubio, Jorge González and Francisco Casacuberta. Bilingual segmentation for phrasetable pruning in Statistical Machine Translation. *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*, 2011. Core B

- Jesús González-Rubio, Daniel Ortiz-Martínez, José-Migel Benedí and Francisco Casacuberta. Interactive Machine Translation using Hierarchical Translation Models. *Proceedings of the conference on Empirical Methods in Natural Language Processing*, 2013. Core A

6.3 Future Work

Finally, we identify the future research directions we intend to explore in a near future to extend the work we have presented.

6.3.1 Machine Translation System Combination

The main research direction to improve the system combination method presented in Chapter 2 involves the study of richer linear loss functions. We have seen that the linear loss BLEU approximation resulted in ill-formed consensus translations. However, it is noticeably more efficient that the other risk function studied: BLEU over expected n-gram counts. We plan to extend the current linear BLEU function by adding new features, such as a language model, that assure the well-formedness of the consensus translations. By doing that, we hope to obtain a new family of linear loss functions that are not only efficient but also match the performance of the complex BLEU risk over expected counts.

Additionally, we plan to release in a near future a public version the MBRSC software used in the experiments on Chapter 2. Not only we consider that MBRSC may be a useful resource for the SMT scientific community, but by releasing a public version of it we hope to gather valuable feedback from the users that surely will allow to further improve the system.

6.3.2 Machine Translation Quality Estimation

There are multiple research directions that we want to explore to improve the QE system proposed in Chapter 3. Initially, we plan to investigate new DR methods. Particularly interesting are the feature selection methods based on minimum-redundancy maximum-relevance [Peng et al., 2005], and the feature extraction methods based in non-linear projections [Lee and Verleysen, 2007]. Another research direction important from the practical point of view is the integration of statistical tests thorough the QE process. These tests will allow us to detect problematic features so that they can be filtered out, and also will help us to analyze and assess the reliability of the results. Finally, we also plan to study various techniques to automatically estimate the optimal size of the reduced set of features, or at least methods that provide a stopping criterion, instead of the manual search currently implemented. These methods have the potential to improve the efficiency of the training process of the QE system.

6.3.3 Active Protocols for IMT

Regarding the active interaction protocols presented in Chapter 4, we plan to investigate more sophisticated, but still fast to compute, confidence measures that improve the accuracy of the predictions. For instance, we can combine different computationally-efficient features by means of a very efficient naïve

Bayes' model. Despite its simplicity, naïve Bayes' models have shown a quite good performance in previous works [Sanchis et al., 2007]. Additionally, we also plan to continue the research on possible approaches to make confidence information available to the user. Specifically, we intend to approach this investigation from two different perspectives. On the one hand, we will carry out investigations on interface design to improving usability of the prototype. On the other hand, given that experiments with human users have shown that some errors are more annoying than others, we plan to modify the evaluation of the future QE models weighting different errors according to how the human users perceive them.

Finally, the main research direction for the active learning framework presented in Chapter 5 will be the efficient implementation of the VOI sampling criteria. This will involve the development of techniques to reduce the set of translations to be explored, techniques to efficiently compute the expected value of supervising an automatic translation, and techniques to efficiently update the model with several alternative sentences pairs. Additionally, we also plan to carry out a human evaluation experiment of the proposed active learning framework for IMT. In this case, the main challenges to be addressed are the evaluation of the actual human cognitive effort, and the management of user variability.

Appendices

Appendix

A IMT Implementation with Word-Graphs

A word-graph is a directed acyclic graph $G = (V, H)$ that can be obtained as a byproduct of the MT search algorithms. It encodes different alternative translation hypotheses in an efficient way. Each vertex $v \in V$ corresponds to a partial translation hypothesis. Each edge $(v, v') \in H$ is annotated with both a target language word $e_{(v,v')}$ and the associated extension probability $p_{(v,v')}$ of language and translation model. The word-graph is constructed in such a way that the extension probabilities only depend on the two adjacent vertexes. So, these probabilities are independent of the considered path through the graph. For simplicity, we assume that there exists exactly one goal and one start node. For a more detailed description of word-graphs, see [Ueffing et al., 2002]. An example of a simplified word-graph for the Spanish source sentence "Transferir documentos explorados a otro directorio" is shown in Figure A.1. The English reference translation is "Move scanned documents to another folder".

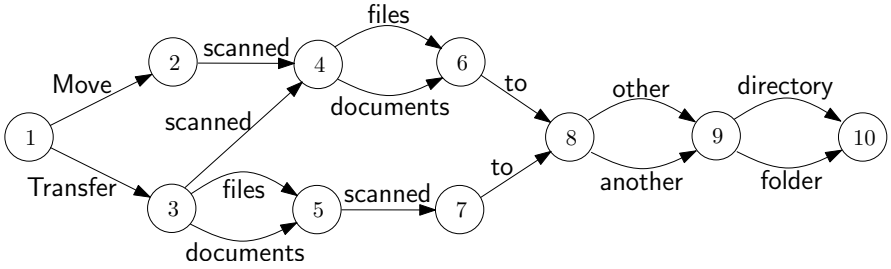


Figure A.1: Example of a word graph for the Spanish source sentence "Transferir documentos explorados a otro directorio".

For each vertex in the word-graph, the maximum probability path to reach the goal node is computed. This probability can be decomposed into the so-called forward probability $\alpha(v)$, which is the maximum probability to reach the vertex v from the start vertex, and the so-called backward probability $\beta(v)$, which is the maximum probability to reach the vertex v backwards from the goal node [Och et al., 2003a].

The backward probability $\beta(v)$ is an optimal heuristic function in the spirit of A* search [Hart et al., 1968]. Having this information, we can compute efficiently for each vertex v in the graph the best successor node $\eta(v)$:

$$\eta(v) = \underset{v':(v,v')\in E}{\arg\max}\ \alpha(v) \cdot p_{(v,v')} \cdot \beta(v) \tag{A.1}$$

As each vertex corresponds to a partial translation hypothesis $\mathbf{e}_p = \mathrm{e}_1 \ldots \mathrm{e}_i$, the optimal extension of this prefix is obtained by:

$$\hat{\mathrm{e}}_{i+1} = e_{(v,\eta(v))} \tag{A.2}$$

$$\hat{\mathrm{e}}_{i+2} = e_{(\eta(v),\eta^2(v))} \tag{A.3}$$

$$\ldots$$

$$\hat{\mathrm{e}}_{i+k} = e_{(\eta^{k-1}(v),\eta^k(v))} \tag{A.4}$$

Hence the function $\eta(\cdot)$ can be used to obtain the optimal word sequence in a time complexity linear to the number of words in the extension.

Yet, as the word-graph contains only a subset of the possible word sequences, we might face the problem that the prefix path is not part of the word-graph. To avoid this problem, an error-tolerant search is usually performed in the word-graph [Och et al., 2003a; Barrachina et al., 2009]. This error-tolerant search starts by selecting the set of vertices with minimum Levenshtein distance to the given prefix. This can be computed by a straightforward extension of the normal Levenshtein algorithm for word-graphs. From this set of vertexes, the one with maximum probability according to Equation (A.1) is chosen and the corresponding extension is computed. Alternatively, Ortiz-Martínez [2011] proposes a IMT formalization that directly includes an stochastic error-correction model in its formulation to address these prefix coverage problems. This alternative formalization is the one used in the IMT results reported in this thesis.

B Linear BLEU derivation

To compute the value of the free parameters $\lambda_0, \lambda_{\mathbf{w}}$ in the linear BLEU definition [Tromble et al., 2008], the authors use a first order Taylor-series approximation to compute what they call the corpus log(BLEU) gain: the change in corpus log(BLEU) contributed by the candidate translation relative to not including that sentence in the corpus.

Let r be the reference length of the corpus, c_0 the candidate translation length, and $\{c_n \mid 1 \leq n \leq 4\}$ the number of n-gram matches. The corpus BLEU score is then defined as[a]:

$$\text{BLEU}(r, c_c, c_n) = \min\left(1, \exp\left(1 - \frac{r}{c_0}\right)\right) \cdot \left(\prod_{n=1}^{4} \frac{c_n}{c_0 - \Delta_n}\right)^{\frac{1}{4}} \tag{B.1}$$

where Δ_n denotes the difference between the number of words in the candidate translation and the number of n-grams. The authors then approximate the log BLEU score from the equation above as follows:

$$\begin{aligned} \log(\text{BLEU}(r, c_0, c_n)) &= \min\left(0, 1 - \frac{r}{c_0}\right) + \frac{1}{4}\sum_{n=1}^{4} \log \frac{c_n}{c_0 - \Delta_n} \\ &\approx \min\left(0, 1 - \frac{r}{c_0}\right) + \frac{1}{4}\sum_{n=1}^{4} \log \frac{c_n}{c_0} \end{aligned} \tag{B.2}$$

where they ignore Δ_n. That is, the authors ignore the n-gram count clippling in their log BLEU approximation.

The corpus log(BLEU) gain G is then defined as the change in log(BLEU) when a new sentence's statistics are added to the corpus statistics:

$$\text{G} = \log(\text{BLEU}(r, c_0', c_n')) - \log(\text{BLEU}(r, c_0, c_n)) \tag{B.3}$$

where the counts c_0', c_n' are equal to c_0, c_n plus the counts of the current sentence. The authors assume that the brevity penalty (first term in Equation (B.2)) does not change when adding the new sentence. They claim that

[a] This definition for BLEU is equivalent to the one provided in Section 1.6.1.

taking into account the brevity penalty at the sentence level cause large performance fluctuations in lattice MBR performance on different test sets. Thereforem they only consider as variables the n-gram matches c_n.

The corpus log(BLEU) gain is then approximated by a first-order vector Taylor series expansion about the initial values of c_n:

$$\mathrm{G} \approx \sum_{n=0}^{N} (c'_n - c_n) \left. \frac{\partial \log(\mathrm{BLEU}(r, c'_0, c'_n))}{\partial c'_n} \right|_{c'_n = c_n} \tag{B.4}$$

where the partial derivatives are given by:

$$\frac{\partial \log(\mathrm{BLEU}(r, c'_0, c'_n))}{\partial c_0} = \frac{-1}{c_0} \qquad \frac{\partial \log(\mathrm{BLEU}(r, c'_0, c'_n))}{\partial c_n} = \frac{1}{4 \cdot c_n} \tag{B.5}$$

Substituting the derivatives in Equation (B.4) gives:

$$\mathrm{G} = \Delta \log(\mathrm{BLEU}) \approx -\frac{\Delta c_0}{c_0} + \frac{1}{4} \sum_{n=1}^{4} \frac{\Delta c_n}{c_n} \tag{B.6}$$

where each $\Delta c_n = c'_n - c_n$ counts the statistic in the sentence of interest. This score is thus a linear function in counts of words Δc_0 and n-gram matches Δc_n.

Using the above first-order approximation to gain in log corpus BLEU, Equation (B.5) imply that $\lambda_0, \lambda_{\mathbf{w}}$ from Equation (2.3.4) would have the following values:

$$\lambda_0 = \frac{-1}{c_0} \qquad \lambda_{\mathbf{w}} = \frac{1}{4 \cdot c_{|\,\mathbf{w}\,|}} \tag{B.7}$$

These factors depend on the length of the current translation c_0 and n-gram matches $(c_n; n \in \{1, 2, 3, 4\})$ that can be obtained from a decoding run on a development set. However, to avoid the dependence on the particular run, the scores are usually estimated making use of the properties of n-gram matches. Since it is known that the average n-gram precisions decay approximately exponentially with n [Papineni et al., 2002], we assume that the number of matches of each n-gram is a constant ratio r times the matches of the corresponding $n-1$ gram. If the 1-gram precision is p, we can obtain the n-gram factors $(\lambda_{\mathbf{w}})$ as a function of the parameters p and r, and the number of 1-gram tokens T:

$$\lambda_0 = \frac{-1}{T} \qquad \lambda_{\mathbf{w}} = \frac{1}{4 \cdot T \cdot p \cdot r^{|\,\mathbf{w}\,|-1}} \tag{B.8}$$

Appendix

C Computation of N-Gram Feature Expectations

The risk functions presented in Chapter 2 use n-gram expectations for its computation. Specifically, these expectations are the expected value for the n-gram indicator features used by linear BLEU risk in Section 2.3.1, and the expected count of each n-gram used by the BLEU risk in Section 2.3.2. Computing these feature expectations from N-best lists of translations is trivial, but N-best lists capture very little of the posterior distribution over translations defined by the SMT model. Different works [Kumar et al., 2009; DeNero et al., 2009, 2010] have shown how these feature expectations can be efficiently computed from more complex representations. Next, we briefly describe how these quantities can be computed from word-graphs or translation forests. Both are acyclic graph-based representations $G = (V, H)$ that efficiently encode multiple translations, thus we use the following convention for both representations: $v \in V$ denote the vertexes in the graph and $h \in H$ denote the edges.

Let $\Phi(\mathbf{e})$ be a vector of features (n-gram indicators or n-gram counts) for a sentence $\mathbf{e}$. Then, $\Phi(\mathbf{e})$ can be computed as the sum of all edge-specific n-gram features $\Phi(h)$ for the edges h in the derivation $\zeta(\mathbf{e})$ that defines $\mathbf{e}$:

$$\mathbb{E}[\Phi(\mathbf{e})] = \sum_{h \in \zeta(\mathbf{e})} \mathbb{E}[\Phi(h)] \tag{C.1}$$

To compute the feature expectations for an edge, we first compute the posterior probability of each edge h, conditioned to the input sentence $\mathbf{f}$:

$$\mathrm{P}(h \mid \mathbf{f}) = \frac{\sum_{\mathbf{e}: h \in \zeta(\mathbf{e})} \mathrm{P}(\mathbf{e} \mid \mathbf{f})}{\sum_{\mathbf{e}} \mathrm{P}(\mathbf{e} \mid \mathbf{f})} \tag{C.2}$$

where the summations iterate over all translations encoded in the selected representation. The numerator can be computed using the forward-backward algorithm for word-graphs, or the inside-outside algorithm for translation forests. The denominator is the forward score of the end node of the word-graph, or respectively the inside score of the root vertex of the translation forest.

The expected n-gram feature vector for an edge is $\mathbb{E}[\Phi(h)] = \mathrm{P}(h \mid \mathbf{f}) \cdot \Phi(h)$. Hence, after computing $\mathrm{P}(h \mid \mathbf{f})$ for every h, the expected value of the features for a translation $\mathbf{e}$ is given by:

$$\mathbb{E}[\Phi(\mathbf{e})] = \sum_{h \in \zeta(e)} \mathrm{P}(h \mid \mathbf{f}) \cdot \Phi(h) \tag{C.3}$$

This entire procedure is a linear-time computation in the number of edges.

Appendix

D On-Line Learning for SMT

The on-line SMT model used to implement the active learning techniques proposed in Chapter 5 was first proposed in [Ortiz-Martínez et al., 2010]. The authors define a log-linear model composed of seven feature functions (from h_1 to h_7) for which a set of sufficient statistics that can be incrementally updated is maintained:

- **n-gram language model**

$$h_1(\mathbf{e}) = \sum_{i=1}^{|\mathbf{e}|+1} \log(\mathrm{P}(\mathrm{e}_i \mid \mathrm{e}_{i-n+1}^{i-1})) \tag{D.1}$$

where $\mathrm{e}_i^j \equiv \mathrm{e}_i \ldots \mathrm{e}_j$, e_0 denotes the begin-of-sentence symbol, $\mathrm{e}_{|\mathbf{e}|+1}$ denotes the end-of-sentence symbol, and $\mathrm{P}(\mathrm{e}_i \mid \mathrm{e}_{i-n+1}^{i-1})$ is defined as follows:

$$\begin{aligned}\mathrm{P}(\mathrm{e}_i \mid \mathrm{e}_{i-n+1}^{i-1}) = &\frac{\max(c_x(\mathrm{e}_{i-n+1}^{i}) - D_n, 0)}{c_x(\mathrm{e}_{i-n+1}^{i-1})} + \\ &\frac{D_n}{c_x(\mathrm{e}_{i-n+1}^{i-1})} \cdot N_{1+}(\mathrm{e}_{i-n+1}^{i-1}\,\bullet) \cdot \mathrm{P}(\mathrm{e}_i \mid \mathrm{e}_{i-n+2}^{i-1})\end{aligned} \tag{D.2}$$

where $D_n = \frac{c_{n,1}}{c_{n,1}+2c_{n,2}}$ is a fixed discount ($c_{n,1}$ and $d_{n,2}$ are the number of n-grams with one and two counts respectively), $N_{1+}(\mathrm{e}_{i-n+1}^{i-1}\,\bullet)$ is the number of unique words that follows the history e_{i-n+1}^{i-1} and $c_x(\mathrm{e}_{i-n+1}^{i})$ is the count of the n-gram e_{i-n+1}^{i}, where $c_x(\cdot)$ can represent true counts $c_{\mathrm{T}}(\cdot)$ or modified counts $c_{\mathrm{M}}(\cdot)$. True counts are used for the higher order n-grams and modified counts for the lower order n-grams. Given a certain n-gram, its modified count consists in the number of different words that precede this n-gram in the training corpus. Equation (D.2) corresponds to an n-gram language model with an interpolated version of the Kneser-Ney smoothing [Chen and Goodman, 1996].

The sufficient statistics for $h_1(\cdot)$ are: $c_{n,1}, c_{n,2}, N_{1+}(\cdot), c_{\mathrm{T}}(\cdot), c_{\mathrm{M}}(\cdot)$

- **Target sentence-length model**

$$h_2(\mathbf{f}, \mathbf{e}) = \log(\phi_{|\,\mathbf{e}\,|}(|\,\mathbf{f}\,| + 0.5)) - \log(\phi_{|\,\mathbf{e}\,|}(|\,\mathbf{f}\,| - 0.5)) \tag{D.3}$$

where $\phi_{|\,\mathbf{e}\,|}(\cdot)$ denotes the cumulative distribution function (cdf) for the normal distribution (the cdf is used to integrate the normal density function over an interval of length one). The authors use a specific normal distribution with mean $\mu_{|\,\mathbf{e}\,|}$ and standard deviation $\sigma_{|\,\mathbf{e}\,|}$ for each possible sentence length $|\,\mathbf{e}\,|$.

The sufficient statistics for $h_2(\cdot)$ are two quantities for each sentence length: $\mu_{|\,\mathbf{e}\,|}$ and $S_{|\,\mathbf{e}\,|}$, where the latter is an auxiliary quantity from which the standard deviation can be computed [Knuth, 1997].

- **Direct and inverse phrase-based models**

$$h_3(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{k=1}^{K} \log(\mathrm{P}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k})) \tag{D.4}$$

where $\mathrm{P}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k})$ is defined as follows:

$$\mathrm{P}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k}) = \beta \cdot \mathrm{P}_{\mathrm{phr}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k}) + (1 - \beta) \cdot \mathrm{P}_{\mathrm{hmm}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k}) \tag{D.5}$$

where each $\tilde{a}_k \in \{1 \ldots K\}$ denotes the index of the target phrase $\tilde{\mathbf{e}}$ that is aligned with the k-th source phrase $\tilde{\mathrm{f}}_k$ assuming a segmentation of length K, $\mathrm{P}_{\mathrm{phr}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k})$ denotes the probability given by a statistical phrase-based dictionary used in regular phrase-based models, and $\mathrm{P}_{\mathrm{hmm}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k})$ is the probability given by a hidden Markov model (HMM) alignment model [Vogel et al., 1996] to each phrase pair. The latter model is used by the authors for smoothing purposes.

Phrase probabilities $\mathrm{P}_{\mathrm{phr}}(\tilde{\mathrm{f}} \mid \tilde{\mathrm{e}})$ are estimated from phrase counts:

$$\mathrm{P}_{\mathrm{phr}}(\tilde{\mathrm{f}} \mid \tilde{\mathrm{e}}) = \frac{c(\tilde{\mathrm{f}}, \tilde{\mathrm{e}})}{\sum_{\tilde{\mathrm{f}}'} c(\tilde{\mathrm{f}}', \tilde{\mathrm{e}})} \tag{D.6}$$

HMM probabilities $\mathrm{P}_{\mathrm{hmm}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k})$ are given by [Vogel et al., 1996]:

$$\mathrm{P}_{\mathrm{hmm}}(\tilde{\mathrm{f}}_k \mid \tilde{\mathrm{e}}_{\tilde{a}_k}) = \sigma \sum_{a_1^{|\tilde{\mathrm{f}}|}} \prod_{j=1}^{|\tilde{\mathrm{f}}|} \mathrm{P}(\tilde{\mathrm{f}}_j \mid \tilde{\mathrm{e}}_{a_j}) \cdot \mathrm{P}(a_j \mid a_{j-1}, |\tilde{\mathrm{e}}|) \tag{D.7}$$

The lexical probability for a pair of words is given by:

$$\mathrm{P}(\mathrm{f} \mid \mathrm{e}) = \frac{c(\mathrm{f} \mid \mathrm{e})}{\sum_{\mathrm{f}'} c(\mathrm{f}' \mid \mathrm{e})} \tag{D.8}$$

where $c(\mathrm{f} \mid \mathrm{e})$ is the expected number of times that the word e is aligned to the word f. The alignment probability is defined in a similar way:

$$\mathrm{P}(a_j \mid a_{j-1}, l) = \frac{c(a_j \mid a_{j-1}, l)}{\sum_{a'_j} c(a'_j \mid a_{j-1}, l)} \tag{D.9}$$

where $c(a_j \mid a_{j-1}, l)$ denotes the expected number of times that the alignment a_j has been seen after the previous alignment a_{j-1} given a source sentence composed of l words.

Analogously h_4 is defined as:

$$h_4(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{k=1}^{K} \log(\mathrm{P}(\tilde{\mathrm{e}}_{\tilde{a}_k} \mid \tilde{\mathrm{f}}_k)) \tag{D.10}$$

The sufficient statistics for $h_3(\cdot)$ and $h_4(\cdot)$ are the phase counts $c(\tilde{\mathrm{f}}, \tilde{\mathrm{e}})$ of the phrase-based model, and the expected word counts $c(\mathrm{f} \mid \mathrm{e})$ and the expected alignment counts $c(a_j \mid a_{j-1}, l)$ of the HMM model.

- **Target phrase-length model**

$$h_5(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{k=1}^{K} \log(\mathrm{P}(|\tilde{\mathrm{e}}_k|)) \tag{D.11}$$

where $\mathrm{P}(|\tilde{\mathrm{e}}_k|) = \delta \cdot (1-\delta)^{|\tilde{\mathrm{e}}_k|}$. This feature implements a target phrase-length model by means of a geometric distribution with a probability of success on each trial equal to δ. The use of a geometric distribution penalizes the length of the target sentences.

This function require none sufficient statistic, the authors left δ fixed.

- **Source phrase-length model**

$$h_6(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{k=1}^{K} \log(\mathrm{P}(|\tilde{\mathrm{f}}_k|, |\tilde{\mathrm{e}}_{\tilde{a}_k}|)) \tag{D.12}$$

where $\mathrm{P}(|\tilde{\mathrm{f}}_k| \mid |\tilde{\mathrm{e}}_{\tilde{a}_k}|) = \frac{1}{1+\tau}\delta(1-\delta)^{\mathrm{abs}(|\tilde{\mathrm{f}}_k| - |\tilde{\mathrm{e}}_{\tilde{a}_k}|)}$, $\tau = \sum_{i=1}^{|\tilde{\mathrm{e}}_{\tilde{a}_k}|-1} \delta(1-\delta)^i$, and $\mathrm{abs}(\cdot)$ is the absolute value function. A geometric distribution (with

scaling factor $\frac{1}{1+\tau}$) is used to model this feature (it penalises the difference between the length of the source and target phrases).

This function require none sufficient statistic, the authors left δ fixed.

- **Distortion model**

$$h_7(\mathbf{a}) = \sum_{k=1}^{K} \log(\mathrm{P}(\tilde{a}_k \mid \tilde{a}_{k-1})) \tag{D.13}$$

where $\mathrm{P}(\tilde{a}_k|\tilde{a}_{k-1}) = \frac{1}{2-\delta}\delta(1-\delta)^{\mathrm{abs}(b_{\tilde{a}_k} - l_{\tilde{a}_{k-1}})}$, $b_{\tilde{a}_k}$ denotes the beginning position of the source phrase covered by $\tilde{a}_k$ and $l_{\tilde{a}_{k-1}}$ denotes the last position of the source phrase covered by $\tilde{a}_{k-1}$. This geometric distribution (with scaling factor $\frac{1}{2-\delta}$) penalizes the reordering between phrases.

This function require none sufficient statistic, the authors left δ fixed.

A detailed description of the algorithms implemented to update each feature from the sufficient statistics stored can be found in [Ortiz-Martínez, 2011].

Appendix

E Symbols and Acronyms

E.1 Mathematical symbols

$\mathbf{x}$	:	Observable object for a classification system
$\mathcal{X}$	:	Domain of observable objects
$\mathbf{y}$	:	Class
$\mathcal{Y}$	:	Domain of classes
$\Pr(\cdot)$	:	General probability distribution with no model assumptions
$\mathrm{P}(\cdot)$	:	Model based probability distribution
$\lvert\cdot\rvert$	:	Cardinality of a set or sequence
$\mathcal{F}$	:	Source language
$\mathbf{f}$	:	Source language sentence
f	:	Source language word
f_j	:	j-th word in a source language sentence
$\mathcal{E}$	:	Target language
$\mathbf{e}$	:	Target language sentence
e	:	Target language word
e_i	:	i-th word in a target language sentence
$\mathbf{e}_p$	:	User validated prefix for a system suggestion $\mathbf{e}$
k	:	User key-stroke
$\mathbf{e}_s$	:	Suffix suggested by the system to complete a user validated prefix $\mathbf{e}_p$
¶	:	Begin of sentence symbol
\$	:	End of sentence symbol
$\mathbf{w}$	:	n-gram, sequence of n consecutive words in a sentence
$\mathcal{W}_n$	:	Set of n-grams of size n in a sentence
$\mathcal{W}$	:	Set of n-grams of all sizes in a sentence
$\mathcal{N}$	:	Multi-set of n-grams in a sentence

E.2 Acronyms

BLEU	:	BiLingual Evaluation Understudy
CAT	:	Computer Assisted Translation
DP	:	Dynamic Programming
DR	:	Dimensionality Reduction
IMT	:	Interactive Machine Translation
KSMR	:	Key-Stroke and Mouse action Ratio
MBR	:	Minimum Bayes' Risk
MBRSC	:	Minimum Bayes' Risk System Combination
MERT	:	Minimum Error Rate Training
MT	:	Machine Translation
NLP	:	Natural Language Processing
PCA	:	Principal Components Analysis
PLSR	:	Partial Least Squares Regression
QE	:	Quality Estimation
RMSE	:	Root Mean Squared Error
SMT	:	Statistical Machine Translation
SVM	:	Support Vector Machines
TER	:	Translation Edit Rate
WSR	:	Word Stroke Ratio

List of Figures

List of Tables

List of Algorithms

Bibliography

S. Abe. *Support Vector Machines for Pattern Classification*. Advances in Pattern Recognition. Springer London, 2010. ISBN 9781849960984.

ALPAC. *Language and machines: computers in translation and linguistics; a report*. Publication (National Research Council (U.S.))). National Academy of Sciences, National Research Council. Automatic Language Processing Advisory Committee, 1966.

E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, Dec. 1998. ISSN 0304-3975.

V. Ambati, S. Vogel, and J. Carbonell. Multi-strategy approaches to active learning for statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*, 2011.

J.-C. Amengual, M. A. Castaño, A. Castellanos, V. M. Jiménez, D. Llorens, A. Marzal, F. Prat, J. M. Vilar, J.-M. Benedí, F. Casacuberta, M. Pastor, and E. Vidal. The eutrans spoken language translation system. *Machine Translation*, 15(1-2):75–103, 2000.

T. W. Anderson. *An Introduction to Multivariate statistical Analysis*. Wiley, New York, 1958.

J. Andrés-Ferrer, D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. On the use of different loss functions in statistical pattern recognition applied to machine translation. *Pattern Recognition Letters*, 29:1072–1081, June 2008. ISSN 0167-8655.

D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, April 1988. ISSN 0885-6125.

L. Atlas, D. Cohn, R. Ladner, M. A. El-Sharkawi, and R. J. Marks, II. Advances in neural information processing systems 2. In D. S. Touretzky, editor, *NIPS*, chapter Training connectionist networks with queries and selective

sampling, pages 566–573. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-100-7. URL http://dl.acm.org/citation.cfm?id=109230.109294.

Atril. Déjà vu x2, 2013. URL http://www.atril.com/en. [Online; accessed 26-March-2013].

E. Avramidis. Quality estimation for machine translation output using linguistic analysis and decoding features. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 84–90. Association for Computational Linguistics, June 2012. URL http://www.aclweb.org/anthology/W12-3108.

J. Aymerich and H. Camelo. The machine translation maturity model at paho. In *Proceedings of the 12th Machine Translation Summit*, pages 403–409, August 26–30 2009.

S. Banerjee and A. Lavie. METEOR: an automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, 2005.

S. Bangalore. Computing consensus translation from multiple machine translation systems. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 351–354, 2001.

S. Bangalore, O. Rambow, and S. Whittaker. Evaluation metrics for generation. In *Proceedings of the first international conference on Natural language generation - Volume 14*, INLG '00, pages 1–8. Association for Computational Linguistics, 2000. ISBN 965-90296-0-8.

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.-M. Vilar. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35: 3–28, March 2009. ISSN 0891-2017.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society*, 35:370–418, 1763.

M. A. Becker. *Active Learning - an Explicit Treatment of Unreliable Parameters*. PhD thesis, University of Edinburgh, 2008.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

R. E. Bellman. *Adaptive control processes: a guided tour*. Rand Corporation Research studies. Princeton University Press, 1961.

O. Bender, S. Hasan, D. Vilar, R. Zens, and H. Ney. Comparison of generation strategies for interactive machine translation. In *Proceedings of the European Association for Machine Translation*, pages 33–40, Budapest, Hungary, May 2005.

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, J. D. Lafferty, R. L. Mercer, H. Printz, and L. Ureš. The candide system for machine translation. In *Proceedings of the workshop on Human Language Technology*, pages 157–162. Association for Computational Linguistics, 1994. ISBN 1-55860-357-3.

A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, March 1996. ISSN 0891-2017.

P. J. Bickel and K. A. Doksum. *Mathematical statistics : basic ideas and selected topics*. Holden-Day, San Francisco, 1977. ISBN 0816207844.

M. Bisani and H. Ney. Bootstrap estimates for confidence intervals in asr performance evaluation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 409–412, May 2004.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. Confidence estimation for machine translation. In *Proceedings of the international conference on Computational Linguistics*, pages 315–321, 2004.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. ISBN 1-58113-057-0. doi: 10.1145/279943.279962.

P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85, 1990. ISSN 0891-2017.

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311, 1993. ISSN 0891-2017.

C. Buck. Black box features for the wmt 2012 quality estimation shared task. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 91–95, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3109.

C. Callison-burch and R. S. Flournoy. A program for automatically selecting the best output from multiple machine translation engines. In *Proceedings of the VIII Machine Translation Summit*, pages 63–66, 2001.

C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. (Meta-) evaluation of machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 136–158, 2007.

C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106. Association for Computational Linguistics, 2008. ISBN 978-1-932432-09-1.

C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, March 2009. Association for Computational Linguistics.

C. Callison-Burch, P. Koehn, C. Monz, and O. F. Zaidan, editors. *Proceedings of the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 2011.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51. Association for Computational Linguistics, June 2012. URL http://statmt.org/wmt12/.

B. A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Transactions on Information Systems*, 30(1):4:1–4:34, Mar. 2012. ISSN 1046-8188.

F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, and S. Molau. Some approaches to

statistical and finite-state speech-to-speech translation. *Computer Speech & Language*, 18(1):25–47, 2004.

F. Casacuberta, J. Civera, E. Cubel, A. L. Lagarda, G. Lapalme, E. Macklovitch, and E. Vidal. Human interaction for high quality machine translation. *Communications of the Association for Computing Machinery*, 52(10):135–138, 2009.

CASMACAT. Cognitive analysis and statistical methods for advanced computer aided translation. Technical Report ICT Project 287576, 2011.

M. A. Castaño and F. Casacuberta. A connectionist approach to machine translation. In *Proceedings of the European Conference on Speech Communication and Technology*, 1997.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Adaptative computation and machine learning series. Mit Press, 2006. ISBN 9780262033589.

K. Chen and H. Chen. A hybrid approach to machine translation system design. *International Journal of Computational Linguistics & Chinese Language Processing*, 1:159–182, August 1996.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/981863.981904.

E. W. Cheney and D. R. Kincaid. *Numerical Mathematics and Computing*. Brooks/Cole, 2012. ISBN 9781133103714.

C. Cherry and G. Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics, 2012. ISBN 978-1-937284-20-6.

N. Chinchor. The statistical significance of the muc-4 results. In *Proceedings of the Conference on Message Understanding*, pages 30–50, 1992. ISBN 1-55860-273-9.

I. Chong and C. Jun. Performance of some variable selection methods when multicollinearity is present. *Chemometrics and Intelligent Laboratory Systems*, 78(1–2):103–112, June 2005.

J. Civera, J. M. Vilar, E. Cubel, A. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó, and J. González. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 207–215. Springer-Verlag, 2004a.

J. Civera, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, and J. González. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the conference on empirical methods for natural language processing*, pages 349–356, Barcelona, Spain, July 25-26 2004b.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994. ISSN 0885-6125.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, Sept. 1995. ISSN 0885-6125.

D. Coughlin. Correlating automated and human assessments of machine translation quality. In *Proceedings of the Machine Translation Summit*, pages 63–70, 2003.

K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Machine Learning Research*, 3:951–991, Mar. 2003. ISSN 1532-4435.

E. Cubel, J. González, A. Lagarda, F. Casacuberta, A. Juan, and E. Vidal. Adapting finite-state translation to the TransType2 project. In *Proceedings of the Joint Conference combining the 8th International Workshop of the European Association for Machine Translation and the 4th Controlled Language Applications Workshop*, pages 54–60, Dublin, Ireland, May 15–17 2003.

E. Cubel, J. Civera, J. M. Vilar, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, J. González, and L. Rodríguez. Finite-state models for computer assisted translation. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 586–590, Valencia, Spain, August 23-27 2004.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. In *The Annals of Mathematical Statistics*, volume 43, pages 1470–1480, 1972.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Proceedings of the international conference on Machine learning*, pages 208–215, 2008. ISBN 978-1-60558-205-4.

N. G. de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie v. Wetenschappen*, 49:758–764, 1946.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.*, 39(1): 1–38, 1977. ISSN 00359246.

J. DeNero, D. Chiang, and K. Knight. Fast consensus decoding over translation forests. In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics*, pages 567–575. Association for Computational Linguistics, 2009. ISBN 978-1-932432-46-6.

J. DeNero, S. Kumar, C. Chelba, and F. Och. Model combination for machine translation. In *Proceedings of the 11th conference of the North American chapter of the Association for Computational Linguistics*, pages 975–983. Association for Computational Linguistics, 2010. ISBN 1-932432-65-5.

T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000. ISBN 3-540-67704-6.

G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

P. Donmez and J. G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 619–628, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3.

N. Duan, M. Li, D. Zhang, and M. Zhou. Mixture model-based minimum bayes risk decoding using multiple machine translation systems. In *Proceedings of the 23rd conference on Computational Linguistics*, pages 313–321, 2010.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001. ISBN 9780471056690.

EC. Translating for a multilingual community. http://ec.europa.eu/dgs/translation/index_en.htm, 2009. European Commision, directorate general for translation.

N. Ehling, R. Zens, and H. Ney. Minimum bayes risk decoding for bleu. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 101–104, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P07-2026.

M. Felice and L. Specia. Linguistic features for quality estimation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 96–103, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3110.

G. Fischer. User modeling in human-computer interaction. *User modeling and user-adapted interaction*, 11(1-2):65–86, Mar. 2001. ISSN 0924-1868. doi: 10.1023/A:1011145532042.

J. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, 1997.

C. Fornell and F. L. Bookstein. Two Structural Equation Models: LISREL and PLS Applied to Consumer Exit-Voice Theory. *Journal of Marketing Research*, 19(4):440–452, 1982.

G. Foster. *Text Prediction for Translators*. PhD thesis, Université de Montréal, may 2002.

G. Foster, P. Isabelle, and P. Plamondon. Target-text mediated interactive machine translation. *Machine Translation*, 12(1/2):175–194, January 1998. ISSN 0922-6567.

G. Foster, P. Langlais, and G. Lapalme. User-friendly text prediction for translators. In *Proceedings of the conference on Empirical methods in natural language processing*, pages 148–155, 2002.

M. Gamon, A. Aue, and M. Smets. Sentence-Level MT evaluation without reference translations: beyond language modeling. In *Proceedings of the conference of the European Association for Machine Translation*. European Association for Machine Translation, 2005.

S. Gandrabur and G. Foster. Confidence estimation for text prediction. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 315–321, 2003.

M. García-Martínez. Selecting translations to be post-edited by sentence-level automatic quality evaluation. Master's thesis, Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, 2012.

G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta. Does more data always yield better translations? In *Proceedings of the European Chapter of the Association for Computational Linguistics*, pages 152–161, 2012.

P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185(1):1–17, Jan. 1986. ISSN 00032670. doi: 10.1016/0003-2670(86)80028-9.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 228–235. Association for Computational Linguistics, 2001.

J. González-Rubio, A. Sanchís, and F. Casacuberta. Prhlt submission to the wmt12 quality estimation task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 104–108. Association for Computational Linguistics, June 2012. URL http://www.aclweb.org/anthology/W12-3111.

W. S. Gosset. The probable error of a mean. *Biometrika*, (1):1–25, 1908. Originally published under the pseudonym "Student".

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 3:1157–1182, Mar. 2003. ISSN 1532-4435.

B. Haddow and P. Koehn. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 422–432, Montreal, Canada, June 2012. Association for Computational Linguistics.

G. Haffari, M. Roy, and A. Sarkar. Active learning for statistical phrase-based machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 415–423, 2009.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009. ISSN 1931-0145.

C. Hardmeier, J. Nivre, and J. Tiedemann. Tree kernels for machine translation quality estimation. In *Proceedings of the* 7^{th} *Workshop on Statistical Machine Translation*, pages 109–113, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3112.

P. Hart, N. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968. doi: 10.1109/TSSC.1968.300136.

S. Hasan, R. Zens, and H. Ney. Are very large n-best lists useful for smt? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 57–60. Association for Computational Linguistics, 2007.

X. He and K. Toutanova. Joint optimization for machine translation system combination. In *Proceedings of the 2009 conference on Empirical Methods in Natural Language Processing*, pages 1202–1211. Association for Computational Linguistics, 2009. ISBN 978-1-932432-63-3.

X. He, M. Yang, J. Gao, P. Nguyen, and R. Moore. Indirect-hmm-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 98–107. Association for Computational Linguistics, 2008.

K. Heafield and A. Lavie. CMU system combination in wmt 2011. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 145–151, Edinburgh, Scotland, 2011. Association for Computational Linguistics.

H. Hotelling. The Generalization of Student's Ratio. *Annals of Mathematical Statistics*, 2(3):360–378, 1931.

J. Hsu. *Multiple Comparisons: Theory and Methods.* Chapman and Hall/CRC, 1996. ISBN 9780412982811.

L. Huang. Advanced dynamic programming in semiring and hypergraph frameworks. In *Coling 2008: Advanced Dynamic Programming in Computational Linguistics: Theory, Algorithms and Applications - Tutorial notes*,

pages 1–18, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL http://www.aclweb.org/anthology/C08-5001.

J. Hutchins. The first public demonstration of machine translation: the Georgetown-IBM system, 7th January 1954. Unpublished, available in http://www.hutchinsweb.me.uk/GU-IBM-2005.pdf, 2005.

W. J. Hutchins and H. L. Somers. *An introduction to machine translation*. Academic Press, 1992. ISBN 978-0-12-362830-5.

IBM. 701 translator. Press release, http://www-03.ibm.com/ibm/history/exhibits/701/701_translator.html, January 8 1954.

P. Isabelle and K. Church. *Special issue on: New tools for human translators*, volume 12. Kluwer Academic Publishers, January 1998.

S. Jayaraman and A. Lavie. Multi-engine machine translation guided by explicit word matching. In *Proceeding of the 10th conference of the European Association for Machine Translation*, pages 143–152, 2005.

F. Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997. ISBN 0-262-10066-5.

A. J. Joshi, F. Porikli, and N. P. Papanikolopoulos. Scalable active learning for multiclass image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2259–2273, 2012.

A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: guiding supervised learning with decision-theoretic active learning. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 877–882, 2007.

M. Kay. The proper place of men and machines inlanguage translation. *Machine Translation*, 12(1/2):3–23, Jan. 1998. ISSN 0922-6567. doi: 10.1023/A:1007911416676.

S. Khadivi and C. Goutte. Tools for corpus alignment and evaluation of the alignments (deliverable d4.9). Technical Report TransType2 (IST-2001-32091), 2003.

J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analisis and Machine Intelligence*, 20:226–239, March 1998. ISSN 0162-8828. doi: 10.1109/34.667881.

K. Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25:607–615, December 1999. ISSN 0891-2017.

K. Knight and Y. Al-Onaizan. Translation with finite-state devices. In *Machine translation and the information soup*, pages 421–437. Springer, 1998.

D. E. Knuth. *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. ISBN 0-201-89684-2.

P. Koehn. *Noun phrase translation*. PhD thesis, Los Angeles, CA, USA, 2003. AAI3133297.

P. Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

P. Koehn and B. Haddow. Interactive assistance to human translators using statistical machine translation methods. In *Proceedings of the the 12th Machine Translation Summit*, Ottawa, Ontario, Canada, August 26-30 2009.

P. Koehn and C. Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121, 2006.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54. Association for Computational Linguistics, 2003.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics, demonstration session*, June 2007.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, Dec. 1997. ISSN 0004-3702.

T. Kohonen. Median strings. *Pattern recognition letters*, 3(5):309–313, 1985.

M. Koponen. Comparing human perceptions of post-editing effort with post-editing operations. In *Proceedings of the Workshop on Statistical Machine*

Translation, pages 181–190, Montreal, Canada, June 2012. Association for Computational Linguistics.

J. V. Kresta, J. F. Macgregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, 69(1):35–47, Feb. 1991. ISSN 00084034.

T. Kristjansson, A. Culotta, P. Viola, and A. McCallum. Interactive information extraction with constrained conditional random fields. In *Proceedings of the 19th national conference on Artifical intelligence*, pages 412–418. AAAI Press, 2004. ISBN 0-262-51183-5.

S. Kumar and W. Byrne. Minimum bayes-risk decoding for statistical machine translation. In D. M. Susan Dumais and S. Roukos, editors, *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 169–176, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics.

S. Kumar, W. Macherey, C. Dyer, and F. Och. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics*, pages 163–171. Association for Computational Linguistics, 2009. ISBN 978-1-932432-45-9.

A. L. Lagarda, V. Alabau, F. Casacuberta, R. Silva, and E. Díaz-de Liaño. Statistical post-editing of a rule-based machine translation system. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 217–220. Association for Computational Linguistics, 2009.

P. Langlais and G. Lapalme. TransType: development-evaluation cycles to boost translator's productivity. *Machine Translation*, 17(2):77–98, September 2002. ISSN 0922-6567.

P. Langlais, G. Foster, and G. Lapalme. TransType: a computer-aided translation typing system. In *Workshop of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Embedded Machine Translation Systems*, EmbedMT '00, pages 46–51. Association for Computational Linguistics, 2000.

P. Langlais, S. Gandrabur, T. Leplus, and L. Guy. The long-term forecast for weather bulletin translation. *Machine Translation*, 19:83–112, March 2005. ISSN 0922-6567.

D. Langlois, S. Raybaud, and K. Smaïli. Loria system for the wmt12 quality estimation shared task. In *Proceedings of the* 7th *Workshop on Statistical Machine Translation*, pages 114–119, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3113.

L. S. Larkey and B. W. Croft. Combining classifiers in text categorization. In H. P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval*, pages 289–297. ACM Press, New York, US, 1996.

J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated, 2007. ISBN 0387393501, 9780387393506.

G. Leusch, M. Freitag, and H. Ney. The RWTH system combination system for wmt 2011. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 152–158, Edinburgh, Scotland, 2011. Association for Computational Linguistics.

V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966. Originally appeared as: В.И. Левенштейн (1965). "Двоичные коды с исправлением выпадений, вставок и замещений символов". Доклады Академий Наук СССР 163 (4): 845–848.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, 1994. ISBN 0-387-19889-X.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220271.

A. Lopez. Statistical machine translation. *ACM Computational Survey*, 40: 8:1–8:49, August 2008. ISSN 0360-0300.

E. Macklovitch. TransType2: the last word. In *Proceedings of the conference on International Language Resources and Evaluation*, pages 167–17, 2006.

L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000. ISSN 0885-2308. doi: 10.1006/csla.2000.0152.

C. Martínez-Hinarejos, A. Juan, and F. Casacuberta. Median strings for k-nearest neighbour classification. *Pattern Recognition Letters*, 24(1–3):173–181, 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(02)00209-X.

E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y. suk Lee, J. B. M. no, M. Paulik, S. Roukos, H. Schwenk, and H. Ney. System combination for machine translation of spoken and written language. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1222–1237, September 2008.

R. G. Miller. *Simultaneous statistical inference*. McGraw-Hill series in probability and statistics. McGraw-Hill, 1966.

E. Moreau and C. Vogel. Quality estimation: an experimental study using unsupervised similarity measures. In *Proceedings of the* 7^{th} *Workshop on Statistical Machine Translation*, pages 120–126, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3114.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pages 355–368, 1999.

J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

L. Nepveu, G. Lapalme, P. Langlais, and G. Foster. Adaptive language and translation models for interactive machine translation. In *Proceedings of the conference on Empirical Methods on Natural Language Processing*, pages 190–197, Barcelona, Spain, July 2004.

H. Ney. Speech translation: coupling of recognition and translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7803-5041-3. doi: 10.1109/ICASSP.1999.758176.

H. Ney, S. Martin, and F. Wessel. Statistical language modeling using leaving-one-out. *Corpus Based Methods in Language and Speech Processing*, pages 174–207, Feb. 1997.

D. V. Nguyen and D. M. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, Jan. 2002. ISSN 1460-2059.

S. Nießen, F. J. Och, G. Leusch, and H. Ney. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 39–45, Athens, Greece, May 2000.

NIST. NIST 2006 machine translation evaluation official results. http://www.itl.nist.gov/iad/mig/tests/mt/, November 2006.

T. Nomoto. Multi-engine machine translation with voted language model. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, pages 494–501. Association for Computational Linguistics, 2004.

E. Noreen. *Computer-intensive methods for testing hypotheses: an introduction*. A Wiley Interscience publication. Wiley, 1989. ISBN 9780471611363.

F. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302. Association for Computational Linguistics, 2002.

F. J. Och, R. Zens, and H. Ney. Efficient search for interactive statistical machine translation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 387–393. Association for Computational Linguistics, 2003a. ISBN 1-333-56789-0.

F. J. Och, R. Zens, and H. Ney. Efficient search for interactive statistical machine translation. In *Proceedings of the European chapter of the Association for Computational Linguistics*, pages 387–393, 2003b. ISBN 1-333-56789-0.

D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11(1):169–198, 1999.

D. Ortiz-Martínez. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation.* PhD thesis, Universitat Politècnica de València, 2011. Advisors: Ismael García Varea and Francisco Casacuberta.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Proceedings of the Machine Translation Summit X*, pages 141–148. Asia-Pacific Association for Machine Translation, Phuket, Thailand, September 2005.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. Online learning for interactive statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 546–554, 2010. ISBN 1-932432-65-5.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318. Association for Computational Linguistics, 2002.

K. A. Papineni, S. Roukos, and T. Ward. Maximum likelihood and discriminative training of direct translation models. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 189–192, May 1998.

A. Patry and P. Langlais. Prediction of words in statistical machine translation using a multilayer perceptron. In *Proceedings of the the 12th Machine Translation Summit*, pages 101–111, Ottawa, Ontario, Canada, 2009.

M. Paul, T. Doi, Y. Hwang, K. Imamura, H. Okuma, and E. Sumita. Nobody is perfect: ATR's hybrid approach to spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 55–62, 2005.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions in Pattern Analysis Machine Intelligence*, 27(8):1226–1238, Aug. 2005. ISSN 0162-8828.

J. C. Platt. Using analytic QP and sparseness to speed training of support vector machines. In *Proceedings of the conference on Advances in neural information processing systems II*, pages 557–563, Cambridge, MA, USA, 1999. MIT Press. ISBN 0-262-11245-0.

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(2): 155–162, January 1964.

R. J. Quinlan. Learning with continuous classes. In *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pages 343–348. World Scientific, 1992.

C. Quirk. Training a sentence-level machine translation confidence measure. In *Proceedings of conference on Language Resources and Evaluation*, pages 825–828, 2004.

V. Romero, A. H. Toselli, and E. Vidal. Character-level interaction in computer-assisted transcription of text images. In *Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition*, pages 539–544, 2010. ISBN 978-0-7695-4221-8. doi: 10.1109/ICFHR.2010.89.

A. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz, and B. Dorr. Combining outputs from multiple machine translation systems. In *Proceedings of the 6th conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235. Association for Computational Linguistics, 2007a.

A. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz. Expected bleu training for graphs: BBN system description for wmt11 system combination task. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 159–165. Association for Computational Linguistics, 2011.

A.-V. Rosti, S. Matsoukas, and R. Schwartz. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, Prague, Czech Republic, June 2007b. Association for Computational Linguistics.

D. Roth and D. Zelenko. Part of speech tagging using a network of linear separators. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 1136–1142. Association for Computational Linguistics, 1998.

N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning*, pages 441–448, 2001. ISBN 1-55860-778-1.

R. Rubino, J. Foster, J. Wagner, J. Roturier, R. Samad Zadeh Kaljahi, and F. Hollowood. Dcu-symantec submission for the wmt 2012 quality estimation task. In *Proceedings of the* 7th *Workshop on Statistical Machine Translation*, pages 138–144, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3117.

A. Sanchis, A. Juan, and E. Vidal. Estimation of confidence measures for machine translation. In *Proceedings of the Machine Translation Summit XI*, pages 407–412, 2007.

R. Schlueter, M. Nussbaum-Thom, and H. Ney. Does the cost function matter in bayes decision rule? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):292–301, 2012.

D. W. Scott and J. R. Thompson. Probability density estimation in higher dimensions. *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, pages 173–179, 1983.

SDL. Sdl trados studio, 2013. URL http://www.trados.com/en/. [Online; accessed 26-March-2013].

N. Serrano, A. Giménez, J. Civera, A. Sanchis, and A. Juan. Interactive handwriting recognition with limited user effort. *International Journal on Document Analysis and Recognition*, pages 1–13, 2013. ISSN 1433-2833. doi: 10.1007/s10032-013-0204-5.

B. Settles. Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin–Madison, 2009.

B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.

C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27:479–523, July 1948.

K. C. Sim, W. J. Byrne, M. J. F. Gales, H. Sahbi, and P. C. Woodland. *Consensus network decoding for statistical machine translation system combination*, volume 4, pages 105–108. IEEE, 2007.

M. Simard and P. Isabelle. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of the 12th Machine Translation Summit*, pages 255–261, Ottawa, Ontario, Canada, 2009.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006.

H. L. Somers. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157, 1999.

H. L. Somers. Chapter 3. In *Computers and translation: a translator's guide*, Benjamins translation library. John Benjamins Publishing Co., 2003. ISBN 9789027216403.

R. Soricut and A. Echihabi. TrustRank: inducing trust in automatic translations via ranking. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 612–621. Association for Computational Linguistics, July 2010.

R. Soricut, N. Bach, and Z. Wang. The sdl language weaver systems in the wmt12 quality estimation shared task. In *Proceedings of the* 7th *Workshop on Statistical Machine Translation*, pages 145–151, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3118.

L. Specia, C. Saunders, Z. Wang, J. Shawe-Taylor, and M. Turchi. Improving the confidence of machine translation quality estimates. In *Proceedings of the Machine Translation Summit XII*, 2009a.

L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. Estimating the sentence-level quality of machine translation systems. In *Proceedins of the meeting of the European Association for Machine Translation*, pages 28–35, 2009b.

R. Stanley. *Enumerative combinatorics*. Cambridge studies in advanced mathematics. Cambridge University Press, 2002. ISBN 9780521663519.

R. Steel and J. Torrie. *Principles and procedures of statistics*. McGraw-Hill, 1960. ISBN 9780070609259.

A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in N-best list rescoring. In G. Kokkinakis, N. Fakotakis, and E. Dermatas,

editors, *Proceedings of the European Conference on Speech Communication and Technology*, volume 1, pages 163–166. ISCA, Rhodes, Greece, 1997.

C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the International Conference on Machine Learning*, pages 406–414, Bled, Slovenia, June 1999.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

P. Toma. Systran as a multilingual machine translation system. In *Proceedings of the Third European Congress on Information Systems and Networks, Overcoming the language barrier*, pages 569–581, 1977.

J. Tomás and F. Casacuberta. Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the Conference on Computational Linguistics*, pages 835–841, Sydney, Australia, 17th-21th July 2006.

R. W. Tromble, S. Kumar, F. Och, and W. Macherey. Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 620–629, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613792.

A. Trujillo. *Translation engines: techniques for machine translation*. Applied computing. Springer, 1999. ISBN 9781852330576.

J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

M. Turchi, T. De Bie, and N. Cristianini. Learning to translate: a statistical and computational analysis. Technical report, University of Bristol, 2009. URL https://patterns.enm.bris.ac.uk/files/LearningCurveMain.pdf.

R. Udupa and H. K. Maji. Computational complexity of statistical machine translation. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–32, 2006. ISBN 1-932432-59-0.

N. Ueffing and H. Ney. Bayes decision rule and confidence measures for statistical machine translation. In *Proceedings of EsTAL - España for Natural Language Processing*, pages 70–81. Springer Verlag, 2004.

N. Ueffing and H. Ney. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the European Association for Machine Translation conference*, pages 262–270, 2005.

N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33:9–40, 2007. ISSN 0891-2017.

N. Ueffing, F. J. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 156–163, 2002.

N. Ueffing, K. Macherey, and H. Ney. Confidence measures for statistical machine translation. In *Proceedings of the MT Summit IX*, pages 394–401. Springer-Verlag, 2003.

B. Vauquois. *La Traduction automatique à Grenoble*. Dunod, 1975. ISBN 9782853440240.

E. Vidal. Finite-state speech-to-speech translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 111–114. IEEE, 1997.

S. Vogel, H. Ney, and C. Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, pages 836–841. Association for Computational Linguistics, 1996.

W. Weaver. Translation. In W. N. Locke and A. D. Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, 1955. Reprinted from a memorandum written by Weaver in 1949.

F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(1):23–31, 2005.

H. Wold. *Estimation of Principal Components and Related Models by Iterative Least squares*, pages 391–420. Academic Press, New York, 1966.

S. Wold, M. Sjöström, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001. ISSN 0169-7439.

D. Xu, Y. Cao, and D. Karakos. Description of the JHU system combination scheme for wmt 2011. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 171–176. Association for Computational Linguistics, 2011.

R. Zens, F. J. Och, H. Ney, and L. F. I. Vi. Phrase-based statistical machine translation. In *Proceedings of Advances in Artificial Intelligence, 25th Annual German Conference on AI*, pages 18–32. Springer Verlag, 2002.

X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from stream data using optimal weight classifier ensemble. *Transactions on Systems Man and Cybernetics Part B*, 40(6):1607–1621, Dec. 2010. ISSN 1083-4419. doi: 10.1109/TSMCB.2010.2042445.

G. K. Zipf. *The Psychobiology of Language*. Houghton-Mifflin, 1935.