Universidad Politécnica de Valencia

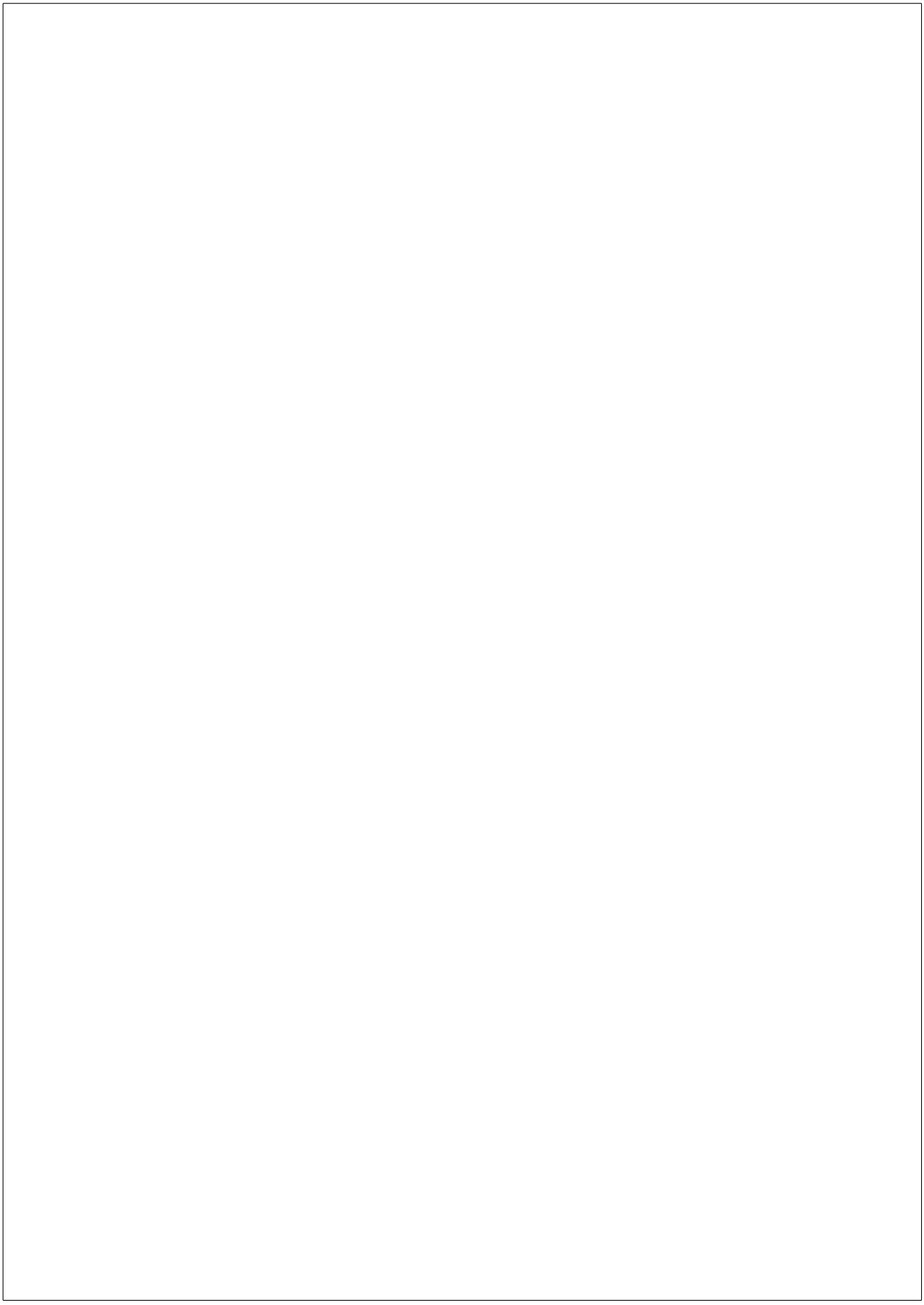
Departamento de Sistemas Informáticos
y Computación

Tesis Doctoral

Aprendizaje de Transductores Estocásticos de Estados Finitos y su Aplicación en Traducción Automática

Autor: Jorge González Mollá
Ingeniero en Informática
Director: Francisco Casacuberta Nolla
Doctor en Ciencias Físicas

Valencia
Julio 2009



Agradecimientos

Este trabajo ha sido respaldado por el Ministerio de Educación de España, bajo la protección del programa de investigación Consolider Ingenio 2010 MIPRCV (CSD2007-00018) y el proyecto iTransDoc (TIN2006-15694-CO2-01).

Esta tesis es el fruto de un trabajo constante y evolutivo que jamás habría visto la luz de no haber aterrizado en un grupo de investigación que me ha tratado desde el principio como a un miembro más de su familia. Desde aquí, mi más sincero agradecimiento a todos sus integrantes, a los actuales y a los extintos, y a toda la gente con la que he tenido el gusto de trabajar durante todo este tiempo. Esto no habría podido ser sin ellos.

Mención especial para mi director de tesis, cuya paciencia para hacer brotar el espíritu investigador en alguien tan perezoso como yo ha sido infinita. Su experiencia y sabios consejos me han introducido en este mundo laboral, que desconocía por completo, logrando la realización personal tan deseada. Paco me ha enseñado a tener autonomía, pero también a trabajar en equipo, a escribir artículos, a documentarlos, en definitiva, a ser un buen profesional. Pero más importante es haberlo hecho sin descuidar nunca el factor humano. Por todo esto y mucho más, mi expresión más adecuada es "¡Gracias!".

Quisiera dedicar esta tesis a todo el equipo de precarios que me ha estado acompañando en mi andadura por este camino. Gente en el ITI o en el DSIC con la que he compartido momentos de asedio y de ocio y que ha contribuido de alguna manera a la realidad de esta tesis. Me gustaría citar a Vicent Alabau, Jesús Andrés, Jorge Civera, Adrián Giménez, Antonio Lagar-

da, Daniel Ortiz, David Picó[1], y Germán Sanchis. También, a Jesús nº 2 y José Ramón, con quienes, junto a algunos anteriores, hicimos posible SisHiTra: lingüística y estados finitos en salsa estadística... ¿a qué me suena todo eso? No quisiera dejar de mencionar a personas como Alejandro, Elsa, Luis, Nico o Jose, con los que también he mantenido aventuras y desventuras en el Poli. A los que no nombro por extensión, deben saber que como integrantes del colectivo, lucharé por todos y cada uno de ellos en cuanto me sea posible para que esta carrera profesional tenga al fin el reconocimiento que tanto merece.

En el terreno personal, a mis padres, nunca les agradeceré suficientemente lo mucho que han creído en mí, lo mucho que me respetan, y su gran amor. Que estas palabras sirvan de agradecimiento a toda una vida de sacrificios. A mi mujer, Beatriz, que me aguanta, me da ánimos, y está soportando con entereza los últimos coletazos de esta tesis, ya por concluir. Parte de este trabajo se lo debo a ella, al permitirme una reclusión casi monacal en la que poder concentrarme al ciento por ciento en la confección de este documento. También quisiera dedicar esta tesis a mi hijo, con sólo unas semanas de vida, como ejemplo de constancia, tesón y voluntad de lo que le espera cuando nazca, si ójala siguiera mis pasos. Pero sobre todo recuerda siempre ser feliz.

A mi hermana, sobrina, amigos, gracias a todos por comprender cómo soy y aceptarme tal cual. Acabo, hace frío en el scriptorium, me duele el pulgar...

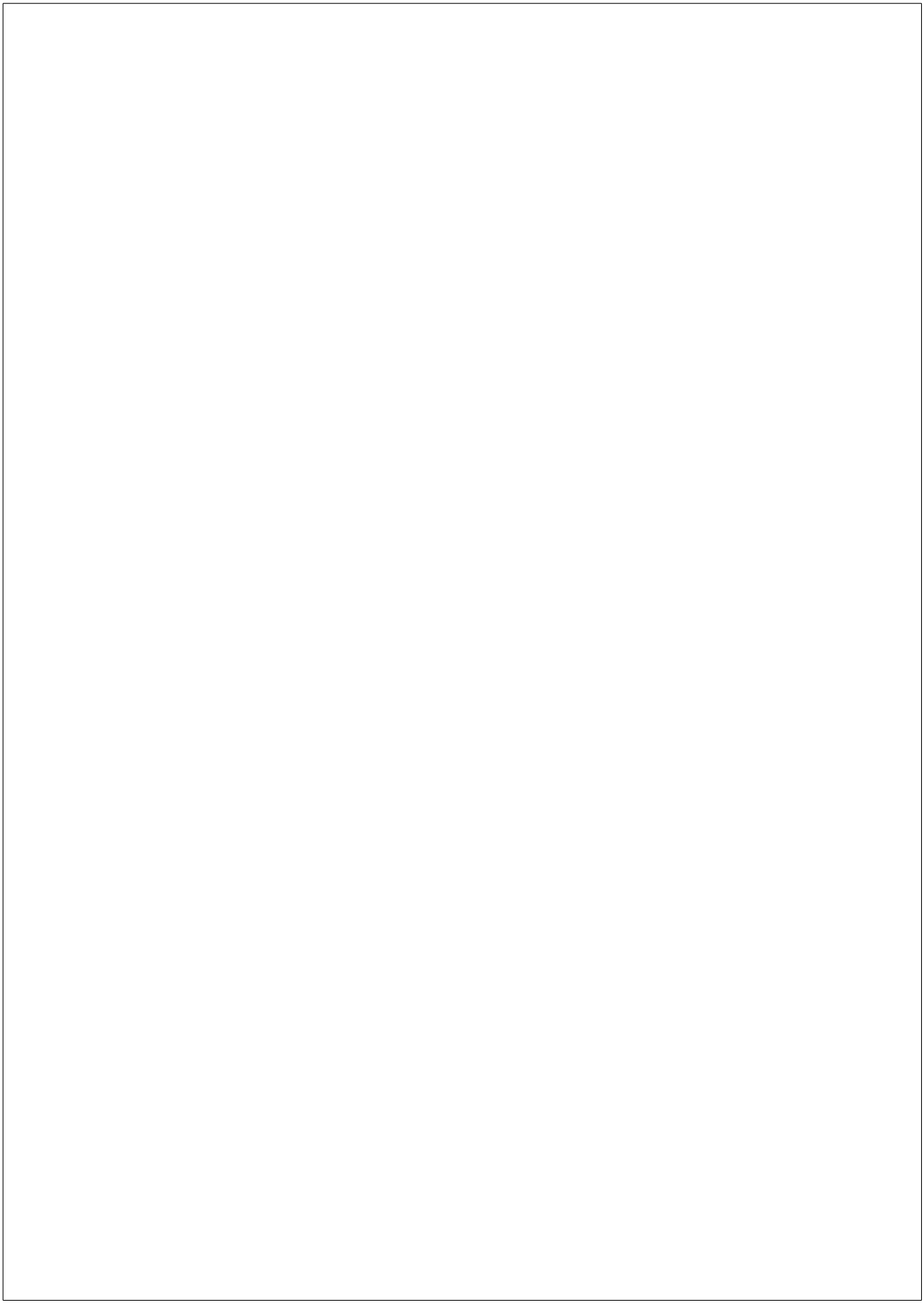
[1] Aunque no he conocido a David en situación precaria, tenía que estar en los agradecimientos

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos



Abstract

Machine translation is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another. In the last decades, there have been a major boost for the use of statistical techniques in the development of machine translation systems. In order to be able to apply these methods on a given language pair, a parallel corpus on those languages needs to be available. These techniques are so attractive because new systems are developed without the need for expert knowledge from linguistic experts.

Finite state models have been very successful in multiple areas from natural language scientific research, machine translation included. Finite state models have some advantages with respect to other statistical models, such as their simple integration into speech recognition environments, their application to computer assisted translation systems, or their ability to process the information, which is not required to be complete, by means of a pipelined architecture that is based on the so popular assembly lines.

The research goal is the study and exploitation of machine translation techniques which are based on finite state models. The work that is presented in this thesis is a detailed analysis about the GIATI methodology on the inference of stochastic finite-state transducers for their effective and efficient application as translation models, allowing their usage on translation tasks with a high volume of data.

On the one hand, a software toolkit that implements the GIATI methodo-

logy in an efficient way has been developed, thus it allows the learning of the structure of the models and the estimation of their probabilities. It also implements different search methods for the evaluation of the models. Moreover, several scalability techniques that allow the usage of a voluminous parallel corpus have been included.

On the other hand, nowadays the state-of-the-art in statistical machine translation is based on the so called phrase-based models. Their inherent idea has been integrated into our framework, allowing the generation of phrase-based transducers which have been contrasted with the ones that are based on words. Their application to GIATI has encouraged an efficient adaptation of the search strategies that has also allowed the usage of more effective smoothing algorithms. We have also incorporated the modern trends on log-linear modelling into this finite-state technology. The approach does a rescoring on transition probabilities in order to increase the system performance.

Finally, the infrastructure for a better exploitation of the available language resources has been established. As a consequence, a better estimation of the translation models would be possible thanks to the use of morphological analizers over the languages which are involved in the translation process. The associated linguistic information allows the application of clustering techniques, thus reducing the corpus variability, then obtaining a statistically more robust model after the training procedure. The experimental results under this approach are rather preliminar but they establish somehow the bases for a future post-doc researching line on this topic.

Resum

Traducció automàtica és un àrea de lingüística computacional que investiga l'ús de software per traduir text o veu en llenguatge natural cap a la seua representació en un idioma destí, també mitjançant llenguatge natural. En les últimes dècades hi ha hagut un fort impuls sobre la utilització de tècniques estadístiques per al desenvolupament de sistemes de traducció automàtica. Per a l'aplicació d'aquests mètodes sobre un parell de llengües en concret, es requereix la disponibilitat d'un corpus paral·lel per a aquest parell d'idiomes. L'atractiu d'aquestes tècniques és que el desenvolupament d'un sistema es fa sense cap necessitat de treball expert per banda d'especialistes en lingüística.

Els models d'estats finits porten prou de temps emprant-se amb èxit en múltiples i variades disciplines dins la investigació científica aplicada al llenguatge natural, incloent el seu ús en traducció automàtica. Els models d'estats finits presenten una sèrie d'avantatges respecte a uns altres models estadístics, com ara la seua senzilla integració en entorns de reconeixement de veu, la seua aplicació en sistemes de traducció assistida, o la seua capacitat per processar la informació sense necessitat de que estiga completa, mitjançant una arquitectura basada en les populars cadenes de muntatge.

L'objectiu de la investigació consisteix en l'estudi i l'explotació de les tècniques de traducció automàtica basades en models d'estats finits. El treball presentat en aquesta tesi és un anàlisi detallat de la metodologia GIATI per a l'aprenentatge de transductors estocàstics d'estats finits per a la seua aplicació eficaç i eficient com a models de traducció, permetent el seu ús sobre

tasques de traducció amb un gran volum de dades.

D'una banda, s'ha desenvolupat un conjunt de ferramentes software que implementen de manera eficient la metodologia GIATI, i que permeten, per tant, l'aprenentatge de l'estructura d'aquests models i l'estimació de les seues probabilitats, incloent mètodes de recerca per a la seua avaluació. A més a més, s'han inclòs diverses tècniques d'escalabilitat en el desenvolupament d'aquestes ferramentes que permeten l'ús d'un corpus paral·lel voluminós.

D'altra banda, en l'actualitat l'estat de l'art en traducció automàtica estadística està basat en els així coneguts models basats en segments. La idea que subjau a aquests models s'ha integrat dins el nostre marc de treball, permetent la construcció de transductors basats en segments, la qualitat dels quals es contrasta positivament front als basats en paraules. La seua aplicació sobre GIATI ha fomentat l'ús d'estratègies de recerca eficients que han permès l'ús d'algorismes de suavitzat més eficaços. També hem adaptat les modernes tendències en modelatge log-lineal sobre aquesta tecnologia basada en transductors estocàstics d'estats finits. L'aproximació permet el refinament de les probabilitats de transició del model, de manera que les prestacions del sistema es veuen incrementades.

Finalment, s'ha establert la infraestructura necessària per a una millor explotació dels recursos lingüístics disponibles. La seua repercussió implica una millor estimació dels models de traducció corresponents, gràcies a l'ús d'analitzadors morfològics en cada llengua implicada en el procés de traducció. La informació lingüística associada permet classificar les paraules en categories, reduint així la variabilitat dels corpus, obtenint uns models estadísticament més robustos després del procés d'aprenentatge. Els resultats mitjançant aquesta aproximació són una mica preliminars però serveixen per establir les bases d'una futura línia d'investigació post-doc sobre aquest tema.

Resumen

Traducción automática es un área de lingüística computacional que investiga el uso de software para traducir texto o voz en lenguaje natural hacia su representación en un idioma destino, también mediante lenguaje natural. En las últimas décadas ha habido un fuerte impulso sobre la utilización de técnicas estadísticas para el desarrollo de sistemas de traducción automática. Para la aplicación de estos métodos sobre un par de lenguas en concreto, se requiere la disponibilidad de un corpus paralelo para dicho par de idiomas. El atractivo de estas técnicas radica en que el desarrollo de un sistema se realiza sin necesidad de trabajo experto por parte de especialistas en lingüística.

Los modelos de estados finitos llevan bastante tiempo empleándose con éxito en múltiples y variadas disciplinas dentro de la investigación científica aplicada al lenguaje natural, incluyendo su uso en traducción automática. Los modelos de estados finitos presentan una serie de ventajas con respecto a otros modelos estadísticos, como una sencilla integración en entornos de reconocimiento de voz, su aplicación a sistemas de traducción asistida, o la capacidad de procesar la información sin necesidad de que esté completa, por medio de una arquitectura basada en las populares cadenas de montaje.

El objetivo de la investigación consiste en el estudio y la explotación de las técnicas de traducción automática basadas en modelos de estados finitos. El trabajo presentado en esta tesis es un análisis detallado de la metodología GIATI para el aprendizaje de transductores estocásticos de estados finitos para su aplicación eficaz y eficiente como modelos estadísticos de traducción,

permitiendo su uso sobre tareas de traducción con un gran volumen de datos.

Por un lado, se ha desarrollado un conjunto de herramientas software que implementan de manera eficiente la metodología GIATI, y que permiten, por tanto, el aprendizaje de la estructura de dichos modelos y la estimación de sus probabilidades, incluyendo métodos de búsqueda para su evaluación. Además, se han incluido varias técnicas de escalabilidad en el desarrollo de dichas herramientas que permiten el uso de un corpus paralelo voluminoso.

Por otro lado, en la actualidad el estado del arte en traducción automática estadística está basado en los así conocidos modelos basados en segmentos. La idea que subyace a estos modelos se ha integrado dentro de nuestro marco de trabajo, permitiendo la construcción de transductores basados en segmentos, cuya calidad se contrasta positivamente frente a los basados en palabras. Su aplicación sobre GIATI ha fomentado el uso de estrategias de búsqueda eficientes que han permitido el uso de algoritmos de suavizado más eficaces. También hemos adaptado las modernas tendencias en modelado log-lineal sobre esta tecnología basada en transductores estocásticos de estados finitos. La aproximación permite el refinamiento de las probabilidades de transición del modelo, de modo que las prestaciones del sistema se ven incrementadas.

Finalmente, se ha establecido la infraestructura necesaria para una mejor explotación de los recursos lingüísticos disponibles. Su repercusión implica una mejor estimación de los modelos de traducción correspondientes, gracias al uso de analizadores morfológicos en cada lengua implicada en el proceso de traducción. La información lingüística asociada permite clasificar las palabras en categorías, reduciendo así la variabilidad de los corpus, obteniendo unos modelos estadísticamente más robustos tras el proceso de aprendizaje. Los resultados por medio de esta aproximación son un tanto preliminares pero sirven para establecer las bases de una futura línea de investigación sobre este tema.

Capítulo 1

Introducción

La mecanización del proceso de traducción ha sido uno de los sueños más antiguos de la humanidad. No en vano, el problema de la coexistencia de diversas lenguas en un mismo ámbito, junto a una más que probable incomprensión entre unos y otros, ya se describió como tal en un libro cuyo origen se remonta a la antigüedad, y que, curiosamente, es el documento al que mayor número de idiomas se ha traducido a lo largo de toda la historia.

No fue hasta el siglo XX que este anhelo se hizo realidad a través de una serie de programas de ordenador, capaces de trasladar de un idioma a otro una gran variedad de textos en lenguaje natural.

Sin embargo, hoy por hoy, la realidad es que estos sistemas de traducción automática no son capaces de producir lo que podríamos llamar traducciones perfectas, por lo que resulta imprescindible una intervención humana para obtener un resultado de calidad, y cuyo origen sea inapreciable para el lector.

Lo cual no debería llevarnos a la conclusión equivocada de que los sistemas de traducción automática son de escasa utilidad debido a la dependencia que mantienen sobre la supervisión constante a la que deben ser sometidos.

Asumiendo que el objetivo del proceso de traducción es de carácter divulgativo, es decir, donde es necesario que el resultado sea de alta calidad,

existen diversas líneas de investigación en las que la productividad de los traductores profesionales se ve incrementada significativamente al combinar su esfuerzo con el apoyo y la asistencia de sistemas de traducción automática.

1.1. Traducción asistida por computador

La gran mayoría de traductores profesionales trabajan en la actualidad para satisfacer la enorme demanda que hay para traducir todo tipo de documentos científico-técnicos, comerciales, administrativos, periodísticos, etc. Parte de este trabajo puede representar un reto, pero mayoritariamente será un proceso tedioso y repetitivo que, sin embargo, requerirá de precisión y consistencia para lograr un producto que cumpla las necesidades del usuario.

El apoyo de un ordenador para este tipo de tareas posee un atractivo claro e inmediato. Sin embargo, la utilidad práctica de un sistema de traducción automática se determinará en último lugar por la calidad de sus traducciones. Si bien es verdad que determinar lo que se considera una *buena* traducción, tanto si se produce por una persona o una máquina, es un concepto extremadamente difícil de definir con precisión. Dependerá de las circunstancias particulares en las que se realice y del destinatario final hacia el que vaya dirigida. Fidelidad, precisión, inteligibilidad, estilo y registro apropiados, son sólo algunos de los criterios que se pueden aplicar, aunque todos ellos son de naturaleza subjetiva. En la práctica, al menos en lo que respecta a los sistemas de traducción automática, lo que verdaderamente importa es cuánto hay que cambiar las hipótesis producidas para convertirlas en traducciones aceptables según las preferencias y las necesidades adecuadas de cada ocasión concreta.

Los sistemas de traducción asistida por computador [IC97] incluyen una amplia variedad de herramientas de ayuda al usuario, entre las que se pueden encontrar correctores ortográficos, gestores terminológicos, diccionarios, memorias de traducción, y también, sistemas de traducción automática. En cuanto a la utilización de estos últimos, el grado de intervención por parte del usuario se mueve entre dos extremos: por un lado, el usuario podría re-

solver corregir a posteriori las traducciones propuestas por el sistema, o en cambio, considerar dichas soluciones como el resultado final de traducción, aceptando las propuestas realizadas por el sistema de traducción automática.

A medio camino entre un funcionamiento autónomo por parte del sistema de traducción automática y una postedición manual de las hipótesis de salida se encuentran los así llamados sistemas de traducción automática interactiva [ELVL04, CCC+08]. En ellos, el motor de traducción se encuentra continuamente en funcionamiento realimentando adecuadamente la hipótesis de salida en función de las modificaciones que el usuario introduce sobre ésta. De este modo, el sistema produce iterativamente una alternativa de traducción completando de manera predictiva la sección que ha validado el usuario.

Este método de traducción predictivo-interactivo se puede implementar de una manera eficaz por medio del uso de transductores de estados finitos. Estos modelos permiten restringir la búsqueda de hipótesis de traducción incorporando con cierta comodidad el concepto de validación de los prefijos. Esta es una de las ventajas de la utilización de transductores de estados finitos en traducción automática, justificando así el desarrollo de esta tesis doctoral.

1.2. Traducción automática a partir de voz

El área de traducción automática a partir de voz presenta una gran relevancia en nuestros días debido en parte al proceso de globalización que está viviendo la humanidad. Tanto desde un punto de vista económico como social, representa uno de los mayores retos intelectuales en el que disciplinas como procesamiento de la señal, reconocimiento del habla, y traducción autómatica, han de cooperar para conseguir el ansiado éxito, coordinando en los últimos tiempos destacadas actividades de investigación en el área [TCS07].

Indudablemente, el proceso de traducción automática a partir de voz presenta una mayor dificultad respecto de los sistemas de traducción de texto. Existe una variable más en el diseño del problema, que consiste en modelar

la voz acústicamente mediante determinados modelos que reflejen adecuadamente la variabilidad existente en el habla, no ya sólo la dependiente del locutor sino también la debida al propio mecanismo biológico que la produce. Ya sea explícitamente, porque así lo requieran las aproximaciones, o bien de un modo implícito, incluido como un subproducto del proceso de traducción, los sistemas de traducción automática a partir de voz permiten descubrir la transcripción textual de la señal vocal pronunciada por el usuario. Por sí sólo, a dicho proceso se le conoce como reconocimiento del habla continua [RJ93].

La tecnología estado del arte en reconocimiento del habla continua está basada en una modelización individual de diversas fuentes de conocimiento, integradas adecuadamente para dar cohesión a la información que modelan. Por un lado, los modelos acústicos suelen modelar información sonora definida generalmente a partir de los fonemas existentes en la lengua escogida. Por otro lado, se encuentran los modelos léxicos y de lenguaje, para modelar respectivamente el vocabulario de la tarea a partir de las unidades fonéticas correspondientes y la generación de frases por medio de dicho vocabulario.

En la actualidad, los modelos acústicos más populares son los así llamados modelos ocultos de Markov (HMMs) [RJ86]. Son un tipo de modelos estadísticos que se caracterizan por la emisión probabilística de secuencias de elementos que pueden ser tanto de naturaleza cualitativa como cuantitativa. Una de las razones de su utilización en reconocimiento del habla es la de que la señal vocal se puede ver como una señal estacionaria en un corto período de tiempo, lo que da pie a modelarla como un proceso estocástico de Markov.

La integración de todas las fuentes de conocimiento para, ante una señal dada, poder obtener su transcripción textual correspondiente, se realiza por medio de la representación de todos los modelos mediante tecnologías de estados finitos, combinando toda la información en un único marco de trabajo que permita la gestión integral de todas las áreas de conocimiento [Vid97]. De esta manera, si sustituimos el modelo de lenguaje por un transductor de estados finitos como modelo de traducción, el sistema de reconocimiento del habla se transforma en un sistema de traducción automática a partir de voz,

que realiza simultáneamente el proceso de reconocimiento y el de traducción.

Dicho de otro modo, los transductores de estados finitos son modelos de traducción que son independientes de una utilización mediante voz o texto, lo que les dota de una atractiva polivalencia frente a otros modelos actuales. Este es otro de los motivos por los que esta tesis explora este tipo de modelos.

1.3. Evaluación en traducción automática

Independientemente de la fuente de entrada de información, la traducción automática en sí es un campo de actividad científico-tecnológica cuya problemática intrínseca impide hoy en día disponer de una solución genérica que resuelva los requerimientos de las sociedades en materia de traducción. Ni siquiera se puede decir que el problema esté completamente resuelto restringiendo el ámbito de actuación sobre un único par de lenguas concretas. En lingüística, existen muchos matices que las máquinas no contemplan aún.

Salvo en tareas de dominio muy limitado (donde quizá las tecnologías actuales de traducción puedan ser efectivas al ciento por ciento), es habitual que los sistemas de traducción automática cometan errores con una cierta frecuencia. De este modo, uno de los problemas más importantes que surgen en traducción automática es el de la evaluación. ¿Cómo evaluar? ¿Qué evaluar? Esas son sólo algunas de las preguntas para las que se han publicado diversas respuestas a lo largo de los años pero que aún continuan abiertas [HKPB02].

Naturalmente, el problema de la evaluación aparece porque generalmente no existe una única traducción que se pueda considerar correcta en su ámbito para una frase de entrada proporcionada. Factores como la sinonimia, flexibilidad en el orden, o simplemente, decir lo mismo pero con otras palabras, son elementos enriquecedores que están presentes en prácticamente todas las lenguas, dificultando la evaluación de los sistemas de traducción automática.

Desde las evaluaciones de carácter subjetivo (en las que factores como la adecuación, la fluidez, etc. [WOO94] pueden inclinar la decisión del eva-

luador hacia un lado de la balanza u otro) hasta las evaluaciones automáticas (en las que generalmente se debe contar con al menos una muestra positiva de una posible solución al problema de traducción propuesto), todas las aproximaciones existentes pretenden establecer un baremo con el que evaluar las hipótesis procedentes de los sistemas de traducción automática, con el fin de obtener un porcentaje de error de los distintos sistemas que permita, no sólo discriminar entre ellos a modo de concurso, sino también indicar en qué medida los sistemas están alcanzando el éxito para el que fueron diseñados.

Ninguna de las dos tendencias (evaluación subjetiva o automática) se encuentra exenta de dificultades. Por un lado, la evaluación subjetiva es muy costosa, ya que involucra a un conjunto de personas, que además deben ser expertas en la materia y que van a necesitar de un tiempo para realizar una correcta calificación. Además, cuando un sistema está en fase de prototipo es habitual que sufra algunos ajustes con el objetivo de mejorar su rendimiento. Variando algunos parámetros del sistema, en ocasiones puede ser necesario obtener distintas evaluaciones en un espacio muy corto de tiempo de modo que la realimentación proporcionada por la evaluación pueda orientar a los diseñadores del sistema acerca de la siguiente variable sobre la que actuar. En dichas situaciones, resulta inabordable acudir a una evaluación subjetiva.

Por otro lado, las evaluaciones automáticas pecan precisamente de ser excesivamente metódicas y de olvidarse de aspectos lingüísticos importantes. Por este motivo existe en la actualidad una corriente importante de investigación que trata de encontrar medidas automáticas con un alto grado de correlación respecto de una evaluación subjetiva [KS04, LRL05, LG07, AH07]. Tanto de si se trata de medidas que ponderan el grado de error del sistema como si lo son respecto de su nivel de aciertos, en ambos casos generalmente hay que calcular dichas tasas respecto de una muestra positiva de traducción. Evidentemente, uno de los problemas con los que una evaluación automática se va a encontrar es el hecho de estar penalizando buenas traducciones simplemente porque no son exactamente como las referencias proporcionadas. Es el precio que se ha de pagar por contar con un medio rápido y económico para disponer de un indicador del rendimiento de un sistema de traducción.

Para tratar de resolver esa dependencia extrema sobre las referencias de traducción, se suele adaptar el cálculo de las medidas automáticas para que puedan operar con multirreferencias, es decir, permitir un conjunto de referencias positivas por cada problema de traducción propuesto, que, de algún modo, contemplen las diferentes soluciones aceptables para dicho problema. Una alternativa a esta posibilidad consiste en evaluar no una única hipótesis sino el conjunto de n-mejores hipótesis, ponderadas mediante ciertos pesos. Si bien es cierto que en ocasiones no existe la posibilidad de disponer de multirreferencias en un corpus, también es frecuente que sea difícil ponderar la jerarquía de n-mejores hipótesis, o que quizá la evaluación de los sistemas no sea pertinente más allá de sus hipótesis más probables, lo que lleva la mayoría de las ocasiones a una evaluación monohipótesis mediante monorreferencia.

En cualquier caso, el conjunto de medidas automáticas presentes en la literatura es vasto y complejo, desde las que estiman tasas de disimilitud, tomando como unidad el carácter, la palabra o el segmento [CBFK+07, CBFK+08], hasta las que valoran el grado de esfuerzo necesario del experto lingüista cuando el sistema se integra en un entorno de traducción interactiva [KPBH03].

En esta tesis, dado el carácter de prototipado del sistema que se ha estado construyendo, se ha escogido para su evaluación un conjunto de medidas automáticas que forman parte del estado del arte en traducción automática, que indican, unas, una estimación del grado de error cometido por el sistema, y otras, a diferencia, una estimación del nivel de éxito logrado por el mismo.

1.4. Modelos de estados finitos

Los modelos de estados finitos llevan bastante tiempo empleándose con éxito en múltiples y variadas disciplinas dentro de la investigación científica aplicada al lenguaje natural, incluyendo su uso en traducción automática. Los modelos de estados finitos presentan una serie de ventajas con respecto a otros modelos estadísticos, como una sencilla integración en entornos de reconocimiento de voz, su aplicación a sistemas de traducción asistida, o la

capacidad de procesar la información sin necesidad de que esté completa, por medio de una arquitectura basada en las populares cadenas de montaje.

Las máquinas de estados finitos son modelos de comportamiento que se componen de un número finito de estados y transiciones entre los mismos. Son modelos abstractos de computación que incorporan una memoria básica, mediante la que cada estado se determina a través de sus estados anteriores. Por este motivo, se dice de ellos que almacenan información sobre el pasado.

Los modelos de estados finitos tienen una relación directa con el álgebra regular, mediante la clase de los lenguajes regulares y las relaciones regulares. A pesar de que existen otros modelos potencialmente más expresivos, los modelos de estados finitos conllevan generalmente un coste computacional menor, lo que les hace más atractivos en la práctica. La literatura describe, además, múltiples desarrollos para su optimización [AM04, MPR02, Moh94].

Por otro lado, las relaciones existentes en traducción son en su mayoría relaciones subsecuenciales, que son un caso particular de relaciones regulares, por lo que, en principio, la capacidad expresiva mediante estados finitos es suficiente para modelar las situaciones que suceden en traducción [Ber79].

Una de las propiedades más interesantes de los modelos de estados finitos es que se pueden inferir automáticamente a partir de una muestra [VGS89]. Sin embargo, el número de aproximaciones existentes varía en gran medida en función del tipo de modelo de estados finitos que se pretenda emplear. Mientras que, por un lado, los métodos para el modelado de lenguajes abundan en la literatura, el número de algoritmos para el aprendizaje de transductores, que modelan relaciones, resulta comparativamente muy limitado. Algunas de estas técnicas son los métodos OSTIA [OGV93] y OMEGA [Vil00]. Otras metodologías, en cambio, modelan el proceso de traducción por medio de una arquitectura de transductores de estados finitos en cascada [KDB06].

Otras aproximaciones para la inferencia de transductores incorporan cierta información estadística que ayude al proceso de aprendizaje [BR01, BR03]. En otros casos, la estimación de modelos se realiza bajo un contexto puramente geométrico, mediante otro tipo de modelos de traducción, a través de los

que, posteriormente, se construyen sus transductores equivalentes [KAO98]. Debido a esto, algunos investigadores consideran que los transductores de estados finitos no son modelos de traducción, propiamente dichos, sino más bien una estructura de datos bajo la que implementar los auténticos modelos.

Por otro lado, en [Mäk99] se introducen una serie de resultados teóricos para poder extender las técnicas de estimación de autómatas a transductores. Esta idea, unida a la anterior, se explota a través del algoritmo GIATI [CVP05] (derivado del inglés *Grammatical Inference Approaches for Transducer Inference*), cuya metodología permite la construcción de transductores estocásticos de estados finitos a través del uso de ciertos algoritmos de inferencia gramatical. Por medio del concepto de segmentación bilingüe monótona, un corpus paralelo se puede transformar en un conjunto de cadenas del que inferir un modelo de lenguaje que se pueda expresar como un autómata finito regular. El alfabeto de estos modelos se compone de símbolos que están formados, a su vez, por símbolos pertenecientes a los idiomas origen y destino objeto del problema de traducción, por lo que la construcción de un transductor a partir del autómata correspondiente se resuelve sustituyendo los símbolos de éste por los respectivos símbolos de entrada y salida de los que están compuestos.

GIATI, el eje sobre el que se desarrolla esta tesis, ha dado lugar a numerosas publicaciones de investigación de importancia relevante dentro del área. No obstante, consideramos que esta técnica se puede explotar más y mejor, no sólo en sí misma, sino también por medio de su combinación con otros modelos, en un marco estadístico más potente como el modelado log-lineal.

1.5. Objetivos de investigación de esta tesis

El objetivo de la investigación consiste en el estudio y la explotación de las técnicas de traducción automática basadas en modelos de estados finitos. Será objeto de análisis la construcción de transductores estocásticos por medio de la metodología GIATI, junto con el proceso de búsqueda de hipótesis de traducción a través de la red estructural que integran este tipo de modelos.

Por un lado, se pretende desarrollar un conjunto de herramientas software que implementen de manera eficiente la metodología GIATI, y que permitan, por tanto, el aprendizaje de la estructura de dichos modelos y la estimación de sus probabilidades, incluyendo métodos de búsqueda para su evaluación. La eficiencia computacional, tanto espacial como temporal, es uno de los problemas que algunas implementaciones previas al desarrollo de esta tesis doctoral presentan de manera evidente [Pic05], haciéndose especialmente patentes cuando el volumen de datos a procesar es de una cierta magnitud. Por tanto, se prevé incluir técnicas de escalabilidad en el desarrollo de dichas herramientas que permitan la utilización de un corpus paralelo voluminoso.

Por otro lado, en la actualidad el estado del arte en traducción automática estadística está basado en los así llamados modelos basados en segmentos. La idea que subyace a estos modelos podría integrarse dentro de nuestro marco de trabajo, permitiendo el desarrollo de un software más completo en sinergia con las tendencias actuales. La adaptación de estas técnicas a nuestra arquitectura basada en estados finitos será otro de los objetivos de esta tesis.

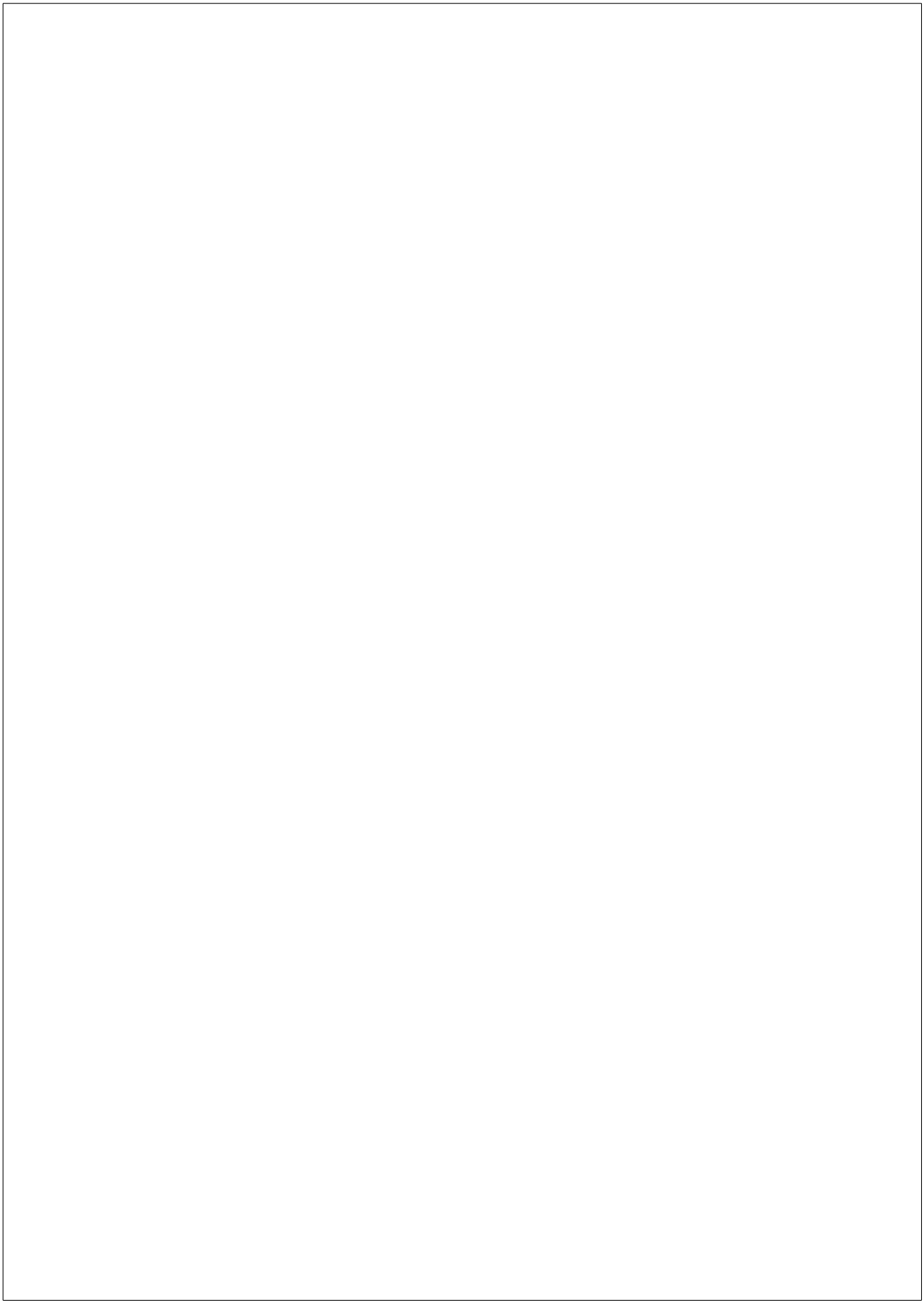
Además, una de las corrientes actuales posibilita la integración de modelos de distinta naturaleza en un único marco estadístico, a partir de una combinación log-lineal. De nuevo, nos proponemos importar esta combinación de modelos a nuestra metodología basada en modelos de estados finitos.

Por último, como objetivo final de esta tesis, pretendemos establecer la infraestructura necesaria para una mejor explotación de los recursos lingüísticos disponibles, que generalmente vienen representados en lenguaje natural. Su repercusión implicaría una mejor estimación de los modelos de traducción correspondientes, gracias al uso de los analizadores morfosintácticos adecuados para cada una de las lenguas implicadas en el proceso de traducción. La información lingüística asociada permitiría clasificar las palabras en categorías, reduciendo de este modo la variabilidad de los corpus, obteniendo unos modelos estadísticamente más robustos tras el proceso de aprendizaje.

1.6. Estructura de este documento de tesis

El trabajo presentado en esta tesis es un estudio detallado de la metodología GIATI para el aprendizaje de transductores estocásticos de estados finitos para su aplicación eficaz y eficiente como modelos estadísticos de traducción, permitendo su uso sobre tareas de traducción con un gran volumen de datos.

Hemos decidido dividir la exposición del documento en varios capítulos. El Capítulo 2 describe los modelos estadísticos de traducción en el contexto de esta tesis. El Capítulo 3, por su parte, explica en detalle la filosofía GIATI para el aprendizaje de transductores estocásticos de estados finitos junto con los algoritmos de descodificación más apropiados para este tipo de modelos. También se incluyen en este capítulo los contenidos relativos a una combinación de modelos bajo la aproximación log-lineal en el marco de GIATI. En el Capítulo 4 se exponen los experimentos relativos a esta metodología conformando los mejores resultados en materia de investigación de la tesis. Por otro lado, el Capítulo 5 describe una posible implementación para la incorporación de información lingüística en este marco de trabajo. Por último, las conclusiones y trabajo futuro de esta tesis se detallan en el Capítulo 6, dando paso a algunos apuntes finales organizados en forma de apéndices junto a la lista de referencias bibliográficas.



Capítulo 2

Traducción automática estadística

Traducción Automática es un campo importante de las tecnologías de la sociedad de la información, integrado en diversos marcos de investigación de la Unión Europea. Dado que el desarrollo de sistemas clásicos de naturaleza deductiva supone un elevado esfuerzo, las aproximaciones estadísticas a la traducción han demostrado ser un marco de trabajo muy interesante debido a su capacidad autónoma para la construcción de sistemas de traducción automática a partir de los correspondientes corpus paralelos [BCDP+90].

Dada una frase de entrada $\mathbf{s}$, el problema de la traducción automática estadística se puede formalizar a través de la siguiente expresión[1]:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}) \tag{2.1}$$

en la que, de entre todas las posibles frases de salida $\mathbf{t}$, $\hat{\mathbf{t}}$ representa a la hipótesis más probable según la distribución de probabilidad condicional a posteriori de $\mathbf{t}$, dada una frase de entrada $\mathbf{s}$, $\Pr(\mathbf{t} \mid \mathbf{s})$. Dicha distribución se aproxima mediante un modelo estadístico $\mathcal{M}$ cuyo aprendizaje se realiza siguiendo un procedimiento basado en técnicas de reconocimiento de formas:

$$\hat{\mathbf{t}} \approx \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}, \mathcal{M}) \tag{2.2}$$

[1] Por simplicidad, $\Pr(X = x)$ se denota mediante $\Pr(x)$, y $\Pr(X = x|Y = y)$, mediante $\Pr(x|y)$

2.1. Modelos de estados finitos

La ecuación 2.2 es la ecuación fundamental en traducción automática estadística, donde $\mathcal{M}$ es un modelo de la distribución de probabilidad $\Pr(\mathbf{t} \mid \mathbf{s})$.

Existe una alternativa que permite el uso de la distribución de probabilidad conjunta $\Pr(\mathbf{s}, \mathbf{t})$ en lugar de $\Pr(\mathbf{t} \mid \mathbf{s})$, a través de la regla que las relaciona:

$$\Pr(\mathbf{t} \mid \mathbf{s}) = \frac{\Pr(\mathbf{s}, \mathbf{t})}{\Pr(\mathbf{s})}$$

y dado que el término del denominador $\Pr(\mathbf{s})$ no depende del factor de maximización $\mathbf{t}$, podemos concluir que el método de búsqueda es independiente de la distribución de probabilidad que modelemos estadísticamente, ya que:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}) = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{s}, \mathbf{t}) \tag{2.3}$$

Aunque existen diferentes aproximaciones para el modelado de una u otra distribución de probabilidad [MW02], ambas se pueden modelar indistintamente mediante transductores estocásticos de estados finitos [CNO+04, CV04]. Estos modelos se comportan de alguna manera como un modelo de traducción y un modelo de lenguaje al mismo tiempo, al tener en cuenta tanto el orden de las palabras como la relación entre los vocabularios de entrada y de salida. Los transductores de estados finitos [CVP05, KDB06] representan un marco de trabajo eficiente en materia de traducción automática, en el que las aplicaciones de traducción en tiempo real se pueden implementar con éxito. Además, los modelos son igualmente válidos para entradas de voz o de texto.

En los siguientes apartados, definimos una serie de modelos que son necesarios para comprender la metodología de traducción basada en modelos de estados finitos que se describe en este documento: autómatas y transductores.

2.1.1. Autómatas de estados finitos

Un autómata de estados finitos se puede definir por medio de una tupla $\mathcal{A} = (\Gamma, Q, I, F, E)$, en la que:

- Γ es un alfabeto de símbolos,
- Q es un conjunto finito de estados,
- $I \subseteq Q$ es un subconjunto de estados iniciales,
- $F \subseteq Q$ es un subconjunto de estados finales, y
- $E \in Q \times \{\Gamma \cup \{\varepsilon\}\} \times Q$ representa un conjunto finito de aristas o transiciones entre estados, en el que cada arista $e = (q_a, \gamma, q_b)$ se caracteriza mediante un estado origen q_a, un símbolo γ, y un estado destino q_b.

Nótese que las aristas pueden ir etiquetadas bien mediante símbolos del alfabeto Γ, o bien mediante la cadena vacía ε. Existen otras definiciones de autómatas en las que no está permitido emplear ε como símbolo de transición, así como alternativas en las que se restringe la cardinalidad del conjunto de estados iniciales y finales a un solo elemento. Sin embargo, se ha demostrado que ninguna de estas variaciones modifica la capacidad de expresión del modelo [AU72]. En la figura 2.1 se puede ver un ejemplo gráfico de autómata.

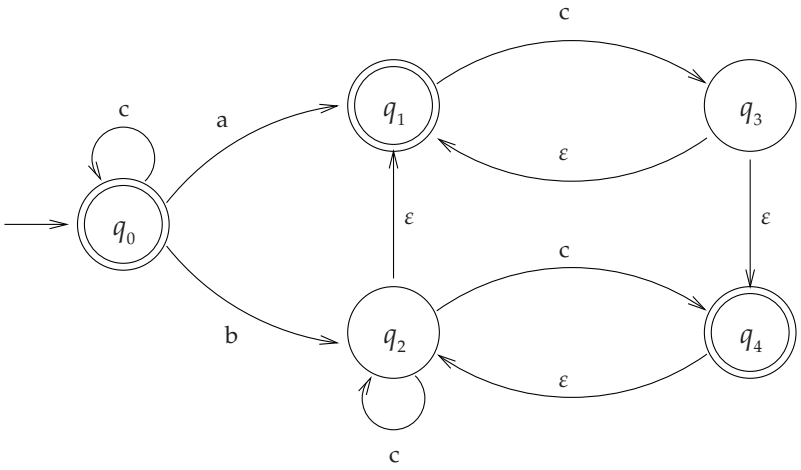


Figura 2.1: Un autómata de estados finitos. Una flecha denota a q_0 como estado inicial; la doble circunferencia señala a q_0, q_1, y q_4 como estados finales.

Se define un camino o derivación π a través de $\mathcal{A}$ como una secuencia de transiciones sucesivas entre los estados del modelo. En particular, se denota mediante $\pi_{q_0,\overline{\gamma},q_m}$ al conjunto de caminos entre los estados q_0 y q_m cuya cadena asociada (la concatenación de los símbolos de sus transiciones) es $\overline{\gamma}$:

$$\pi_{q_0,\overline{\gamma},q_m} = \{(q_0,\gamma_1,q_1)(q_1,\gamma_2,q_2)\dots(q_{m-1},\gamma_m,q_m)\}$$

de tal manera que $\overline{\gamma} = \gamma_1\gamma_2\dots\gamma_m$, y $\forall i = 1\dots m$, entonces $(q_{i-1},\gamma_i,q_i) \in E$.

Se dice que un autómata es propio si no contiene estados o aristas inútiles, es decir, que todos sus estados y aristas forman parte de, al menos, un camino que comienza en un estado inicial del autómata y concluye en un estado final.

El conjunto de caminos entre un estado inicial y un estado final del modelo, y cuya cadena asociada sea $\overline{\gamma}$, se indicará mediante la expresión $d(\overline{\gamma})$:

$$d(\overline{\gamma}) = \{\pi_{q,\overline{\gamma},q'} : q \in I, q' \in F\}$$

De esta manera, el lenguaje asociado a un autómata se puede definir a partir de la expresión anterior como el conjunto de aquellas cadenas para las que existe al menos un camino entre estados iniciales y finales del modelo:

$$L(\mathcal{A}) = \{\overline{\gamma} \in \Gamma^\star : d(\overline{\gamma}) \neq \emptyset\} \tag{2.4}$$

Si $d(\overline{\gamma})$ es un conjunto que incluye varios caminos, quiere decir que existen diversas formas de analizar la cadena $\overline{\gamma}$ a través del modelo, por lo que se dice que el autómata $\mathcal{A}$ presenta ambigüedad con respecto a dicha cadena. De este modo, para considerar que un autómata no es ambiguo se tiene que cumplir que no exista ninguna cadena de $L(\mathcal{A})$ para la que el modelo presente ambigüedad, es decir, todas las cadenas se modelan con un único camino:

$$\forall\overline{\gamma} \in L(\mathcal{A}) : |d(\overline{\gamma})| = 1$$

El autómata de la figura 2.1 no cumple esta propiedad, luego es ambiguo, porque existe al menos una cadena ("ac") que está asociada a varios caminos.

Una condición suficiente para que un autómata no sea ambiguo es que sea determinista, es decir, que todos los estados del modelo cumplen que sus

transiciones de salida se etiquetan con símbolos que no se repiten entre sí. Por ejemplo, el modelo de la figura 2.1 es no determinista (o indeterminista) ya que el estado q_3 presenta ambigüedad de transición respecto al símbolo ε.

La familia de lenguajes que se pueden modelar mediante autómatas de estados finitos es la familia de los lenguajes regulares.

Para profundizar sobre la teoría de lenguajes y autómatas, véase [HMU06].

2.1.2. Autómatas estocásticos de estados finitos

La extensión estocástica de los autómatas de estados finitos permite la incorporación del concepto de probabilidad asociada a un camino, estableciendo distribuciones de probabilidad sobre los lenguajes que modelan [Paz71].

Un autómata estocástico de estados finitos es una tupla $\mathcal{A} = (\Gamma, Q, i, f, P)$, en la que Γ es un alfabeto de símbolos, Q es un conjunto finito de estados, las funciones $i : Q \to [0,1]$ y $f : Q \to [0,1]$ se refieren a la probabilidad de cada estado de ser, respectivamente, un estado inicial y un estado final, y la función $P : Q \times \{\Gamma \cup \{\varepsilon\}\} \times Q \to [0,1]$ define un conjunto de transiciones entre estados de tal manera que cada transición se etiqueta mediante un símbolo de Γ (o mediante la cadena vacía ε), y a la que se le asigna una probabilidad.

En función de si definimos el proceso estocástico bien como un modelo generativo (en la línea de lo que se conoce como gramáticas regulares estocásticas [Wet80, CO94]), bien como un modelo aceptor [Rab63, Fu82], las restricciones sobre la consistencia de las funciones varían. En el caso generativo, las funciones i, P, y f, representan ciertas probabilidades condicionales sobre los posibles movimientos del modelo, (inicio, transición, y finalización), cuya expresión se realiza exclusivamente mediante operaciones de transición, con la ayuda de dos estados virtuales, INI y FIN, que simulan ser los únicos estados de inicio y finalización, respectivamente, y de dos símbolos auxilia-

res, $<s>$ y $</s>$, como símbolos de inicio y fin de cadena, respectivamente:

$$\begin{aligned} i(q) &= \Pr(q, <s> | INI) \\ P(q, \gamma, q') &= \Pr(q', \gamma | q) \\ f(q) &= \Pr(FIN, </s> | q) \end{aligned}$$

de modo que $\forall \gamma \neq <s> : \Pr(q, \gamma | INI) = 0$ y $\forall \gamma \neq </s> : \Pr(FIN, \gamma | q) = 0$. Por tanto, es necesario garantizar que se cumplen las siguientes condiciones:

$$\sum_{q \in Q} i(q) = 1$$

$$\forall q \in Q : \sum_{\gamma \in \{\Gamma \cup \{\varepsilon\}\}, q' \in Q} P(q, \gamma, q') + f(q) = 1$$

En cambio, si consideramos el proceso estocástico como un modelo aceptor, las distribuciones de probabilidad asociadas a las funciones i, f y P son:

$$\begin{aligned} i(q) &= \Pr(q | INI, <s>) \\ P(q, \gamma, q') &= \Pr(q' | q, \gamma) \\ f(q) &= \Pr(FIN | q, </s>) \end{aligned}$$

por consiguiente, las condiciones que se deben respetar para garantizar una distribución de probabilidad consistente sobre las cadenas del autómata son:

$$\sum_{q \in Q} i(q) = 1$$

$$\forall q \in Q, \gamma \in \{\Gamma \cup \{\varepsilon\}\} \wedge (q, \gamma, q') \in P : \sum_{q' \in Q} P(q, \gamma, q') = 1$$

$$\forall q \in Q : f(q) = 0 \ \vee \ f(q) = 1$$

donde las aristas inexistentes del autómata se deben representar formalmente como transiciones hacia un nuevo estado virtual de topología absorvente, permitiendo así completar totalmente la consistencia del modelo estocástico.

Según sea la naturaleza del problema a resolver, convendrá modelar el proceso estocástico de una manera o de otra, teniendo en cuenta las diferencias existentes entre ambas aproximaciones. Básicamente, un modelo generativo reparte la masa de probabilidad entre todas las decisiones disponibles a partir de un cierto estado dado, mientras que un modelo aceptor decreta un reparto entre todas las decisiones que afecten al mismo símbolo de transición. En la figura 2.2 se puede observar un autómata estocástico de estados finitos que ha sido construido siguiendo un modelo estocástico del tipo generativo.

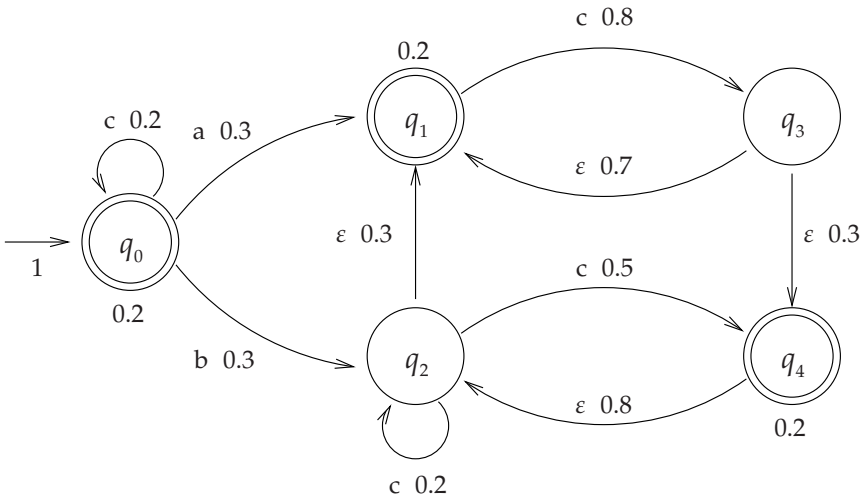


Figura 2.2: Un autómata estocástico definido como un modelo generativo. Como se aprecia, las decisiones disponibles para cualquier estado suman 1.

El autómata característico de un autómata estocástico es el autómata de estados finitos que resulta de eliminar las probabilidades del modelo, es decir, es aquel que comparte el alfabeto y el conjunto de estados, considera como estados iniciales a aquellos cuya probabilidad de ser inicial es superior a 0, del mismo modo, asigna al conjunto de estados finales a aquellos estados cuya probabilidad de ser estado final es superior a 0, y, finalmente, tiene en cuenta el conjunto de aristas cuya probabilidad de transición es distinta de 0.

Un modelo estocástico aceptor presenta una cierta utilidad respecto a su autómata característico siempre que el modelo sea indeterminista, es decir, que presente ambigüedad de transición frente a los símbolos del alfabeto, puesto que, de otro modo, todas las probabilidades de transición valdrían 1, y el modelo estocástico se comportaría de forma equivalente al característico. En la figura 2.3 se puede observar un autómata estocástico de estados finitos que ha sido construido por medio de un proceso estocástico del tipo aceptor.

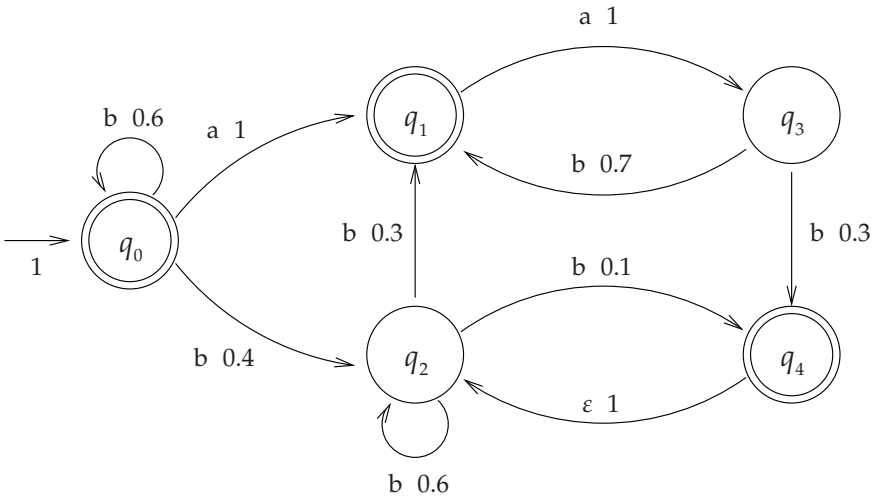


Figura 2.3: Un autómata estocástico definido como un modelo aceptor

Se define la probabilidad de un camino $\pi \in \pi_{q_0,\overline{\gamma},q_m}$ a través de $\mathcal{A}$ como el productorio de las probabilidades de las transiciones de las que se compone:

$$\Pr_{\mathcal{A}}(\pi : \pi \in \pi_{q_0,\overline{\gamma},q_m}) = \prod_{i=1}^{m} P(q_{i-1}, \gamma_i, q_i)$$

De este modo, la probabilidad asignada por el autómata estocástico $\mathcal{A}$ sobre una cierta cadena $\overline{\gamma}$, lo que se denotará mediante la función $F_{\mathcal{A}}(\overline{\gamma})$, es:

$$F_{\mathcal{A}}(\overline{\gamma}) = \sum_{\pi \in \pi_{q,\overline{\gamma},q'}} i(q) \Pr_{\mathcal{A}}(\pi) f(q') \tag{2.5}$$

Por ejemplo, la generación de la cadena "ac" en el modelo de la figura 2.2 se puede realizar por medio de tres rutas alternativas. Uno de estos caminos es $(q_0, a, q_1)(q_1, c, q_3)(q_3, \varepsilon, q_4)$, cuya contribución sobre $F_{\mathcal{A}}(ac)$ es de 0.0144.

El lenguaje regular estocástico que define el modelo $\mathcal{A}$ es el par $(L(\mathcal{A}), F_{\mathcal{A}})$, donde $L(\mathcal{A})$ es el lenguaje regular asociado al autómata característico de $\mathcal{A}$.

Los autómatas característicos de sendos autómatas estocásticos, construidos a partir de cualquier modelo, bien sea éste de tipo generativo o aceptor, son iguales, y por consiguiente modelan el mismo conjunto de cadenas $L(\mathcal{A})$. Sin embargo, en función de si el proceso estocástico se define por medio de un modelo de tipo generativo o aceptor, las propiedades sobre $F_{\mathcal{A}}$ variarán, ya que la distribución de probabilidad que modela cada proceso es diferente.

Por ejemplo, en el caso generativo, y siempre que el autómata sea propio (subsección 2.1.1), la distribución que modela el autómata es $\Pr(\overline{\gamma})$, por lo que la masa de probabilidad se reparte completamente entre todas las cadenas de $L(\mathcal{A})$, representando fielmente el concepto asociado a un modelo de lenguaje:

$$F_{\mathcal{A}}(\overline{\gamma}) \approx \Pr(\overline{\gamma})$$

$$\sum_{\overline{\gamma} \in \Gamma^\star} F_{\mathcal{A}}(\overline{\gamma}) = 1$$

Sin embargo, en el caso aceptor, un autómata sin aristas-ε modela la probabilidad de aceptar o no las cadenas. Así, asumiendo que el modelo es propio, que todos los estados transitan al menos una vez con cada símbolo del alfabeto, y que también son estados finales, el lenguaje estocástico definido a partir de las probabilidades que el modelo asigna a las cadenas del lenguaje no resulta un modelo más descriptivo que su autómata característico, ya que:

$$F_{\mathcal{A}}(\overline{\gamma}) \approx \Pr(ACEPTAR \mid \overline{\gamma})$$

$$F_{\mathcal{A}}(\overline{\gamma}) = \begin{cases} 1 & \forall \overline{\gamma} \in L(\mathcal{A}) \\ 0 & \forall \overline{\gamma} \notin L(\mathcal{A}) \end{cases}$$

$$\sum_{\overline{\gamma} \in \Gamma^\star} F_{\mathcal{A}}(\overline{\gamma}) = |L(\mathcal{A})|$$

debido a que cada cadena $\overline{\gamma} \in L(\mathcal{A})$ integra su propia distribución de probabilidad, cuya masa se reparte entre todas las derivaciones que contiene $d(\overline{\gamma})$. Si alguna de las condiciones anteriores no se cumple, la masa de probabilidad no se distribuye completamente, de manera que $\forall \overline{\gamma} \in L(\mathcal{A}) : F_{\mathcal{A}}(\overline{\gamma}) \leq 1$, luego el sumatorio presenta una cota superior: $\sum_{\overline{\gamma} \in \Gamma^\star} F_{\mathcal{A}}(\overline{\gamma}) \leq |L(\mathcal{A})|$ [Cas90]. En cualquier caso, dicho sumatorio carece de importancia en este contexto, cuyo valor podría ser el de infinito si $L(\mathcal{A})$ es un lenguaje de tal naturaleza.

2.1.3. Transductores de estados finitos

Un transductor de estados finitos se puede definir por medio de una tupla $\mathcal{T} = (\Sigma, \Delta, Q, I, F, E)$, en la que:

- Σ es un alfabeto de símbolos de entrada,
- Δ es un alfabeto de símbolos de salida,
- Q es un conjunto finito de estados,
- $I \subseteq Q$ es un subconjunto de estados iniciales,
- $F \subseteq Q$ es un subconjunto de estados finales, y
- $E \subseteq Q \times \{\Sigma \cup \{\varepsilon\}\} \times \Delta^\star \times Q$ representa un conjunto finito de aristas o transiciones entre estados, en el que ahora cada arista $e = (q_a, s, \bar{t}, q_b)$ se caracteriza mediante un estado origen q_a, un símbolo de entrada s, una cadena de salida $\bar{t}$, y un estado destino q_b.

Una definición más genérica para los transductores de estados finitos permite describir las transiciones del modelo usando cadenas de símbolos de entrada, evitanto tener que restringir su utilización a símbolos individuales. Sin embargo, y debido a que ambas aproximaciones presentan la misma capacidad de expresión, utilizaremos indistintamente una definición u otra en aras de la comprensión de los algoritmos descritos en el ámbito de esta tesis.

Nótese asimismo que las aristas se pueden etiquetar mediante la cadena vacía tanto en la entrada como en la salida, e incluso en ambos lados a la vez. En la figura 2.4 se esboza un ejemplo visual de transductor de estados finitos.

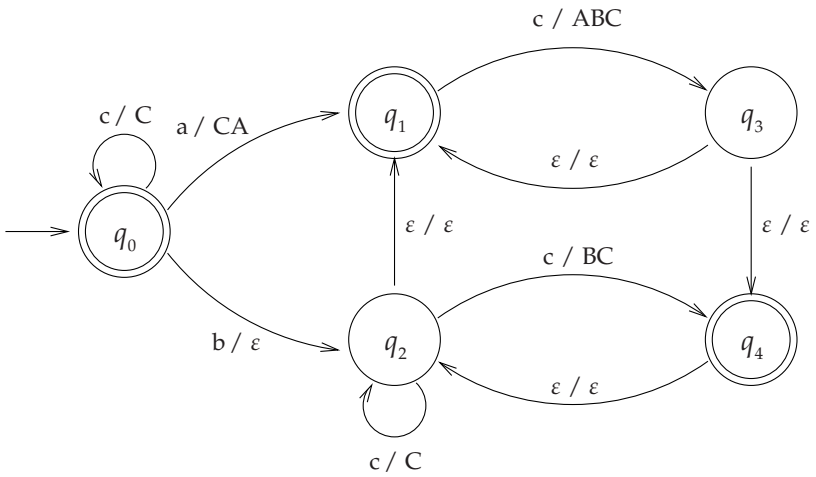


Figura 2.4: Un transductor de estados finitos. Los caracteres en minúsculas simbolizan el alfabeto de entrada. Los caracteres en mayúsculas, el de salida.

De manera similar a como se hizo con los autómatas, se denota mediante $\pi_{q_0,\bar{s},\bar{t},q_m}$ al conjunto de derivaciones a través de $\mathcal{T}$, entre los estados q_0 y q_m, y cuyas cadenas asociadas de entrada y de salida son, respectivamente, $\bar{s}$ y $\bar{t}$.

Asimismo, se considera un transductor como propio si no contiene elementos inútiles, es decir, que todos sus estados y aristas forman parte de, al menos, un camino entre un estado inicial y un estado final del transductor.

De igual modo, el conjunto de caminos entre un estado inicial y un estado final del modelo, cuyas cadenas asociadas de entrada y de salida sean $\mathbf{s}$ y $\mathbf{t}$, se expresará mediante el uso de la función d: $d(\mathbf{s},\mathbf{t}) = \{\pi_{q,\mathbf{s},\mathbf{t},q'} : q \in I, q' \in F\}$.

Así, la relación asociada a un transductor $\mathcal{T}$ se define como el conjunto de aquellos pares de cadenas para los que existe al menos un camino entre estados iniciales y finales de $\mathcal{T}$: $R(\mathcal{T}) = \{(\mathbf{s},\mathbf{t}) \in \Sigma^\star \times \Delta^\star : d(\mathbf{s},\mathbf{t}) \neq \emptyset\}$.

A diferencia de los autómatas, la ambigüedad de un transductor se puede definir en tres ámbitos diferentes, que son: respecto a las cadenas de entrada, con respecto a las cadenas de salida, y con respecto a ambas cadenas a la vez. Esta última forma de ambigüedad se puede expresar de forma parecida a como se hizo con los autómatas, es decir, si $d(\mathbf{s}, \mathbf{t})$ es un conjunto que incluye varios caminos, significa que existen diversas formas de relacionar el par de cadenas $\mathbf{s}$ y $\mathbf{t}$ a través del modelo, por lo que se dice que el transductor $\mathcal{T}$ presenta ambigüedad gramatical respecto del citado par de entrada y salida. En cambio, para definir el concepto de ambigüedad respecto de una cadena aislada, bien sea de entrada o de salida, lo que hay que averiguar es la cantidad de relaciones en las que interviene dicha cadena en $R(\mathcal{T})$, de modo que si este número es superior a uno, el transductor es ambiguo respecto de ella.

Así, para considerar que un transductor no es gramaticalmente ambiguo se tiene que cumplir que no exista ningún par de cadenas de $R(\mathcal{T})$ cuyo conjunto de caminos que las relacionan tenga una cardinalidad superior a 1:

$$\forall(\mathbf{s}, \mathbf{t}) \in R(\mathcal{T}) : |d(\mathbf{s}, \mathbf{t})| = 1$$

Por otro lado, un transductor no es ambiguo respecto de la entrada si $\forall(\mathbf{s}, \mathbf{t}) \in R(\mathcal{T}) : \neg\exists\, \mathbf{t}' \neq \mathbf{t} \wedge (\mathbf{s}, \mathbf{t}') \in R(\mathcal{T})$. Igualmente, un transductor no es ambiguo respecto de la salida si $\forall(\mathbf{s}, \mathbf{t}) \in R(\mathcal{T}) : \neg\exists\, \mathbf{s}' \neq \mathbf{s} \wedge (\mathbf{s}', \mathbf{t}) \in R(\mathcal{T})$. Nótese que ninguno de estos tres tipos de ambigüedad implica a los demás.

Sin embargo, la definición de los transductores de estados finitos como una extensión a partir de los autómatas de estados finitos no sólo obedece a términos expresivos sino también a los teoremas que los relacionan [CVP05] y que dan lugar a los métodos de inferencia de transductores del Capítulo 3. A grandes rasgos, lo que vienen a demostrar es que para todo transductor $\mathcal{T}$, existe un autómata $\mathcal{A}$, cuyas cadenas $\overline{\gamma} \in L(\mathcal{A}) \subseteq \Gamma^\star$, aplicadas sobre sendos homomorfismos $h_\Sigma : \Gamma^\star \to \Sigma^\star$ y $h_\Delta : \Gamma^\star \to \Delta^\star$, producen los pares de cadenas $(h_\Sigma(\overline{\gamma}), h_\Delta(\overline{\gamma})) \in \Sigma^\star \times \Delta^\star$ que pertenecen a la relación del transductor $R(\mathcal{T})$.

La familia de relaciones que se pueden modelar mediante transductores de estados finitos es la familia de las relaciones regulares.

2.1.4. Transductores estocásticos de estados finitos

La extensión estocástica para los transductores es similar a la descrita para los autómatas. Así, un transductor estocástico de estados finitos se define del mismo modo que un autómata estocástico de estados finitos, salvo por el hecho de que las transiciones entre estados se etiquetan mediante símbolos que pertenecen a dos alfabetos (un alfabeto de entrada y un alfabeto de salida). La inserción de un proceso estocástico en los transductores de estados finitos permite asociar probabilidades a pares de cadenas de entrada/salida [CV04].

Un transductor estocástico de estados finitos es una tupla $\mathcal{T} = (\Sigma, \Delta, Q, i, f, P)$, en la que Σ y Δ son los alfabetos de entrada y de salida respectivamente, Q es un conjunto finito de estados, las funciones $i : Q \to [0,1]$ y $f : Q \to [0,1]$ se refieren a la probabilidad de cada estado de ser, respectivamente, un estado inicial y un estado final, y la función $P : Q \times \{\Sigma \cup \{\varepsilon\}\} \times \Delta^\star \times Q \to [0,1]$ define un conjunto de transiciones entre pares de estados de tal manera que cada transición se etiqueta mediante un símbolo de entrada (o la cadena vacía ε), se le asocia una cadena de salida, y a la que se le asigna cierta probabilidad.

En función de si definimos el proceso estocástico bien como un modelo generativo, bien como un modelo aceptor, las restricciones sobre la consistencia de las funciones varían. En el caso generativo, las distribuciones de probabilidad que modelan las funciones i, P, y f, están asociadas a los posibles movimientos del modelo, expresados como transiciones entre estados, mediante la simulación del inicio y el final a través de dos estados virtuales, de manera similar al planteamiento definido para los autómatas estocásticos:

$$\begin{aligned} i(q) &= \Pr(q, <s>, \varepsilon \,|INI) \\ P(q, s, \bar{t}, q') &= \Pr(q', s, \bar{t}|q) \\ f(q) &= \Pr(FIN, </s>, \varepsilon \,|q) \end{aligned}$$

con $\forall s \neq <s>, \bar{t} \neq \varepsilon : \Pr(q, s, \bar{t}|INI) = 0$ y $\forall s \neq </s>, \bar{t} \neq \varepsilon : \Pr(FIN, s, \bar{t}|q) = 0$. Por tanto, es necesario garantizar que se cumplen las siguientes condiciones:

$$\sum_{q \in Q} i(q) = 1$$

$$\forall q \in Q : \sum_{s \in \{\Sigma \cup \{\varepsilon\}\}, \bar{t} \in \Delta^\star, q' \in Q} P(q, s, \bar{t}, q') + f(q) = 1$$

En cambio, si consideramos el proceso estocástico como un modelo aceptor, las distribuciones de probabilidad asociadas a las funciones de $\mathcal{T}$ serán:

$$\begin{aligned} i(q) &= \Pr(q, \varepsilon | INI, <s>) \\ P(q, s, \bar{t}, q') &= \Pr(q', \bar{t} | q, s) \\ f(q) &= \Pr(FIN, \varepsilon | q, </s>) \end{aligned}$$

donde $\forall \bar{t} \neq \varepsilon : \Pr(q, \bar{t} | INI, <s>) = 0$ y $\forall \bar{t} \neq \varepsilon : \Pr(FIN, \bar{t} | q, </s>) = 0$, luego las condiciones que se deben respetar para garantizar una distribución consistente de probabilidades sobre la relación asociada al transductor $\mathcal{T}$ son:

$$\sum_{q \in Q} i(q) = 1$$

$$\forall q \in Q, s \in \{\Sigma \cup \{\varepsilon\}\} \wedge (q, s, \bar{t}, q') \in P : \sum_{\bar{t} \in \Delta^\star, q' \in Q} P(q, s, \bar{t}, q') = 1$$

$$\forall q \in Q : f(q) = 0 \ \vee \ f(q) = 1$$

Nuevamente, los símbolos de entrada sin transición a partir de cierto estado forman el conjunto de aristas a generar hacia un estado virtual sumidero, permitiendo así completar totalmente la consistencia del modelo estocástico.

En la figura 2.5 se puede observar un transductor estocástico de estados finitos que se ha construido por medio de un proceso estocástico generativo. En la figura 2.6 se puede observar un transductor estocástico de estados finitos que ha sido generado mediante un modelo estocástico de tipo aceptor.

El transductor característico de un transductor estocástico es el transductor de estados finitos que resulta de eliminar las probabilidades del modelo.

Se define la probabilidad de un camino $\pi \in \pi_{q_0, \bar{s}, \bar{t}, q_m}$ a través de $\mathcal{T}$ como el productorio de las probabilidades de las transiciones de las que se compone:

$$\Pr_{\mathcal{T}}(\pi : \pi \in \pi_{q_0, \bar{s}, \bar{t}, q_m}) = \prod_{i=1}^{m} P(q_{i-1}, \bar{s}_i, \bar{t}_i, q_i)$$

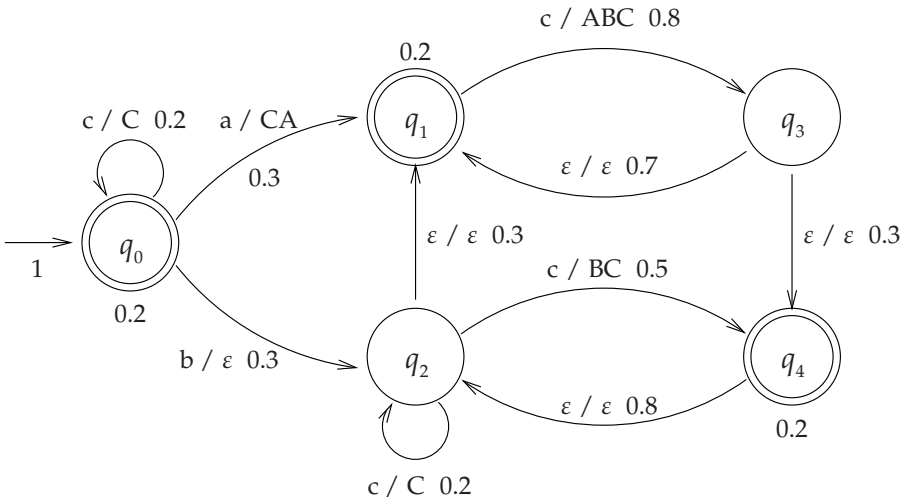


Figura 2.5: Un transductor estocástico definido como un modelo generativo

Por lo tanto, al igual que con los autómatas, dada una cadena de entrada y una de salida, la probabilidad asociada a dichas cadenas por el modelo será:

$$F_{\mathcal{T}}(\mathbf{s},\mathbf{t}) = \sum_{\pi \in \pi_{q,\mathbf{s},\mathbf{t},q'}} i(q)\Pr_{\mathcal{T}}(\pi)f(q') \tag{2.6}$$

La relación regular estocástica que define el modelo $\mathcal{T}$ es el par $(R(\mathcal{T}), F_{\mathcal{T}})$ donde $R(\mathcal{T})$ es la relación regular asociada al transductor característico de $\mathcal{T}$.

Nuevamente, en función de si el proceso estocástico se define por medio de un modelo generativo o de un modelo aceptor, las propiedades sobre $F_{\mathcal{T}}$ variarán, debido a que la probabilidad que modela cada proceso es diferente.

Por ejemplo, bajo un proceso generativo, la distribución de probabilidad que modela el transductor es la conjunta, por lo que la masa de probabilidad se reparte entre todos los pares de cadenas que pertenecen a la relación $R(\mathcal{T})$:

$$F_{\mathcal{T}}(\mathbf{s},\mathbf{t}) \approx \Pr(\mathbf{s},\mathbf{t})$$

$$\sum_{\mathbf{s}\in\Sigma^\star,\mathbf{t}\in\Delta^\star} F_{\mathcal{T}}(\mathbf{s},\mathbf{t}) = 1$$

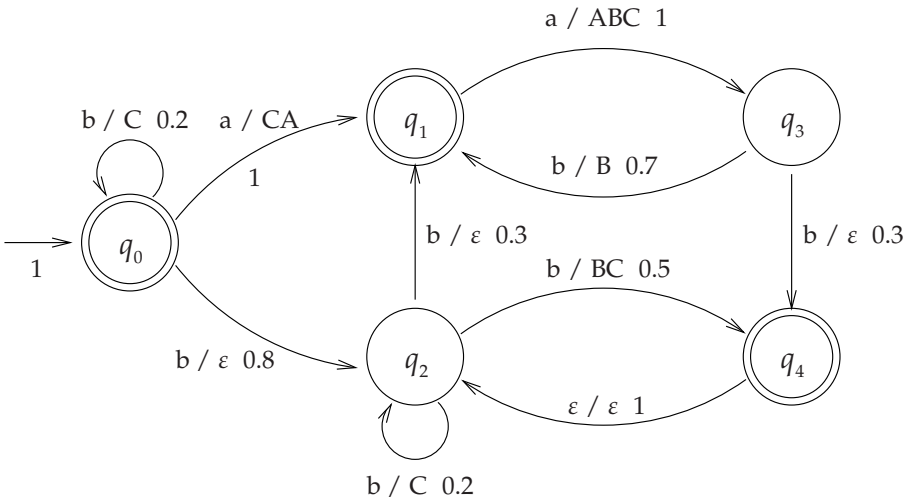


Figura 2.6: Un transductor estocástico definido como un modelo aceptor

Sin embargo, en el caso aceptor, la probabilidad que modela el transductor es la condicional $\Pr(\mathbf{t} \mid \mathbf{s})$, por lo que existe una distribución de probabilidad por cada cadena de entrada, lo que permite un reparto probabilístico condicional entre todas las cadenas de salida con las que se relaciona según $R(\mathcal{T})$:

$$F_{\mathcal{T}}(\mathbf{s}, \mathbf{t}) \approx \Pr(\mathbf{t} \mid \mathbf{s})$$

$$\forall\, \mathbf{s} \in \Sigma^{\star} \wedge (\mathbf{s}, \mathbf{t}) \in R(\mathcal{T}) : \sum_{\mathbf{t} \in \Delta^{\star}} F_{\mathcal{T}}(\mathbf{s}, \mathbf{t}) = 1$$

En ambos casos, se ha de cumplir que el transductor no presenta estados o aristas inútiles para que la(s) distribución(es) de probabilidad repartan por completo toda la masa de probabilidad entre los distintos eventos del modelo. Para el caso aceptor, es necesario además que todos los estados transiten al menos una vez con cada símbolo del alfabeto de entrada (pero no con ε), y ya que el proceso de finalización se modela como una transición especial, tambien hace falta que todos los estados del transductor sean estados finales.

2.1.5. Búsqueda a través de modelos de estados finitos

La utilización más interesante de los transductores estocásticos de estados finitos surge en el momento que, dada una cadena de entrada $\mathbf{s} \in \Sigma^\star$, queremos obtener su traducción en un determinado lenguaje de salida, es decir, cuando buscamos aquella cadena $\hat{\mathbf{t}} \in \Delta^\star$ cuya relación con $\mathbf{s}$, $(\mathbf{s}, \hat{\mathbf{t}}) \in R(\mathcal{T})$, es la más probable según el modelo estadístico con respecto al resto de relaciones en las que interviene $\mathbf{s}$ como cadena de entrada: $\hat{\mathbf{t}} = \underset{\mathbf{t} \in \Delta^\star}{\operatorname{argmax}} F_{\mathcal{T}}(\mathbf{s}, \mathbf{t})$.

Indudablemente, el problema se puede resolver mucho más fácilmente en el caso de que el transductor sea determinista con respecto a la entrada, ya que la solución es la cadena de salida con la que únicamente se relaciona. De hecho, el transductor característico resuelve el problema de igual forma. Sin embargo, excepto en los experimentos diseñados expresamente ad-hoc, la ambigüedad tiende a ser una característica presente en el lenguaje natural, dado lo cual, las probabilidades pueden servir de ayuda para superar la barrera de la ambigüedad y ser capaces así de tomar una decisión consecuente.

No obstante, se ha demostrado que la resolución de cualquiera de las ecuaciones de (2.3) utilizando modelos de estados finitos es un problema computacional NP duro [CdlH00], cuya única solución algorítmica consiste en una variación del algoritmo $A^\star$, de coste exponencial, por lo que en la práctica se implementa con ayuda de ciertos heurísticos para reducir el coste.

Sin embargo, podemos emplear una aproximación a la ecuación (2.6) por medio de la contribución de mayor peso en el sumatorio de dicha ecuación:

$$F_{\mathcal{T}}(\mathbf{s}, \mathbf{t}) \approx \max_{\pi \in \pi_{q,\mathbf{s},\mathbf{t},q'}} i(q) \Pr_{\mathcal{T}}(\pi) f(q') \tag{2.7}$$

de manera que la probabilidad asociada a un par de cadenas según el modelo se aproxima mediante la probabilidad asociada a su derivación más probable.

De este modo, el cálculo de las expresiones de (2.3) se convierte en un proceso de menor coste computacional, con solución algorítmica en tiempo polinómico [Cas95, PC01], a través del algoritmo de análisis de Viterbi [Vit67].

Algoritmo de Viterbi

El algoritmo de Viterbi fue inicialmente desarrollado para encontrar, dada una cadena de símbolos, la secuencia de estados más probable para producir dicha cadena mediante un modelo de Markov [For73]. Este problema es equivalente al de la búsqueda en transductores estocásticos de estados finitos.

El análisis de una cadena de entrada a través de un transductor estocástico de estados finitos se basa en un problema resoluble mediante programación dinámica: la búsqueda del camino de coste mínimo en un grafo multietapa.

Se define un grafo dirigido $G = (V, A)$ como un conjunto V de vértices (también llamados nodos) y un conjunto $A = \{(v, w) : v \in V, w \in V\}$ de aristas, en el que cada arista tiene asociada una función de coste $P : A \rightarrow \mathbb{R}$.

Un grafo multietapa es un grafo dirigido en el que:

- El conjunto de vértices está particionado en J etapas.

$$V = \bigcup_{j=1}^{J} V_j \qquad J \geq 2, \forall j, j' = 1, \ldots, J \wedge j \neq j' : V_j \cap V_{j'} = \emptyset$$

- Las aristas conectan sucesivamente los vértices entre etapas adyacentes.

$$\forall (v, w) \in A : \exists j, 2 \leq j \leq J : (v \in V_{j-1}) \wedge (w \in V_j)$$

- Existe un conjunto de vértices iniciales $I \subseteq V_1$ y vértices finales $F \subseteq V_J$.

Esta es la definición estándar de un grafo multietapa. Sin embargo, dado que los modelos introducidos en las secciones 2.1.3 y 2.1.4 incluyen la posibilidad de transitar mediante la cadena vacía, el alcance de las aristas se amplía permitiendo que puedan conectar también dos vértices de una misma etapa:

$$\forall (v, w) \in A : \exists j, 2 \leq j \leq J : (v \in V_{j-1} \cup V_j) \wedge (w \in V_j)$$

Un subgrafo $g(q)$ de un grafo multietapa G es un grafo que contiene todos los caminos en G que van desde cualquier vértice inicial hasta el nodo q,

ubicado en una cierta etapa. Al conjunto de vértices de dos etapas adyacentes que se conectan con un arco al vértice q se le denota mediante el conjunto W_q:

$$W_q = \{w \in V_{j-1} \cup V_j : (w, q) \in A\} \qquad 2 \leq j \leq J, \;\; q \in V_j$$

El camino de coste mínimo a través de un grafo multietapa G recorre el grafo desde un vértice inicial hasta un vértice final, optimizando una función de acumulación C que se basa en una cierta operación matemática $\otimes$. Gráficamente, la figura 2.7 es un ejemplo. Para resolver este problema por programación dinámica, se puede utilizar la siguiente relación de recurrencia:

$$C(q) = \underset{w \in W_q}{\text{optimiza}} \left[C(w) \otimes P(w, q) \right] \tag{2.8}$$

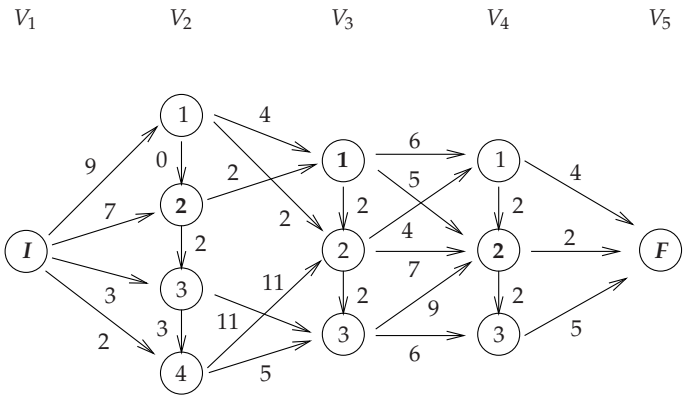


Figura 2.7: Un grafo de 5 etapas [HS78]. La primera y la última, con un vértice inicial y uno final. Su camino de coste mínimo asociado es la secuencia de estados $\boldsymbol{I - 2 - 1 - 2 - F}$, con un coste acumulado de 16 unidades, lo que se determina mediante una minimización basada en el operador de la suma.

Cada problema $C(q)$ asociado al subgrafo $g(q)$ se descompone en los subproblemas relativos a los vértices $w \in W_q$ junto con la aportación particular de la función de peso $P(w, q)$ de la arista que conecta dichos vértices. El número de posibles decisiones en cada punto dependerá del número de

arcos entrantes en cada vértice. Si N es el número medio de aristas entrantes por nodo, entonces la complejidad espacial del algoritmo en su versión iterativa será $\Theta(|V|)$, mientras que su complejidad temporal será $\Theta(|V| \cdot N)$.

Dado un transductor estocástico de estados finitos $\mathcal{T}$ y una cadena de entrada $\mathbf{s}$, existe un grafo multietapa, conocido mediante la voz inglesa *trellis*, de manera que la resolución de las expresiones de (2.3) se reduce a la búsqueda del camino de coste mínimo (o de máxima probabilidad) en dicho grafo.

Sea $\mathcal{T} = (\Sigma, \Delta, Q, i, f, P)$ un transductor estocástico de estados finitos, y sea $\mathbf{s} = \mathbf{s}_1 \, \mathbf{s}_2 \ldots \mathbf{s}_J$ una cadena de entrada de longitud J. El *trellis* o grafo dirigido multietapa asociado $G = (V, A)$ se definirá de la siguiente manera:

- Cada etapa tiene tantos vértices como estados tiene el transductor $\mathcal{T}$. Hay una etapa por cada símbolo de la cadena, más una etapa inicial V_0.

$$V = \bigcup_{j=0}^{J} V_j \qquad V_j = \{q : q \in Q\}$$

- Habrá un arco entre los vértices $v \in V_{j-1}$ y $w \in V_j$ si y sólo si existe una transición entre los estados v y w de $\mathcal{T}$ cuyo símbolo de entrada sea $\mathbf{s}_j$.

  Asimismo, habrá un arco entre dos nodos de la misma etapa si y sólo si existe una transición entre ellos en $\mathcal{T}$ con ε como símbolo de entrada.

$$A = \bigcup_{j=1}^{J} A_j$$

$$A_j = \left\{ \begin{array}{llll} (v,w) : v \in V_{j-1} & , \quad w \in V_j & \wedge & \exists \bar{t} \in \Delta^\star : (v, \mathbf{s}_j, \bar{t}, w) \in P \\ (v,w) : v \in V_j & , \quad w \in V_j & \wedge & \exists \bar{t} \in \Delta^\star : (v, \varepsilon, \bar{t}, w) \in P \end{array} \right\}$$

- El conjunto de nodos iniciales está formado por los estados del transductor cuya probabilidad de ser estado inicial del modelo sea no nula.

$$I \subseteq V_0 = \{q \in Q : i(q) > 0\}$$

- El conjunto de vértices finales está constituido por aquellos vértices de la última etapa que se corresponden con estados finales del transductor.

$$F \subseteq V_J = \{q \in Q : f(q) > 0\}$$

- Los costes de los arcos del *trellis* provienen directamente de las probabilidades de transición del transductor estocástico de estados finitos $\mathcal{T}$.

Dada la aproximación introducida en la ecuación (2.7), la relación de recurrencia (2.8) se instancia a través de la definición de sus propios parámetros:

$$\begin{aligned} \otimes &= \text{producto} \\ \text{optimiza} &= \text{maximiza} \end{aligned}$$

Por lo que la resolución mediante programación dinámica es la siguiente:

$$C(q) = \max_{w \in W_q} [C(w) \cdot P(w, q)]$$

Para calcular la expresión (2.7), sólo hay que realizar una iteración más desde los vértices finales del *trellis*, añadiendo la probabilidad de ser estado final según el transductor estocástico $\mathcal{T}$ como coste asociado a dichas aristas:

$$F_{\mathcal{T}}(\mathbf{s}, \hat{\mathbf{t}}) \approx \max_{q \in F} [C(q) \cdot f(q)]$$

En su versión iterativa, la complejidad espacial de este algoritmo, que depende del número de vértices del *trellis*, es de $\Theta(J \cdot |Q|)$. Si el número medio de arcos por vértice es N la complejidad temporal es de $\Theta(J \cdot |Q| \cdot N)$. Sin embargo, el coste espacial se puede reducir a $\Theta(|Q|)$ teniendo en cuenta que en cada etapa sólo se requieren los valores asociados a los nodos de dicha etapa (y de la inmediata anterior), pudiendo despreciar los de etapas previas. Por tanto, en lugar de emplear la matriz completa, se puede usar únicamente dos vectores columna para representar sendas etapas consecutivas del *trellis*.

El algoritmo de Viterbi, en su versión iterativa, se describe a continuación:

Algoritmo 2.1 Versión iterativa del algoritmo de Viterbi

Datos

$\mathbf{s} = \mathbf{s}_1\, \mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$

$\mathcal{T} = (\Sigma, \Delta, Q, i, f, P)$

Resultado $F_{\mathcal{T}}(\mathbf{s}, \hat{\mathbf{t}}) : \mathbb{R}$

Variables $C_1^{|Q|}, C_1'^{|Q|} : \mathbb{R}$

Método

$\forall q \in I$

$C'_q = i(q)$

$\forall j = 1 \ldots J$

$\forall q \in Q$

$C_q = \max_{w \in W_q} [\ C'_w \cdot P(w, q)\ ]$

$C' = C$

$F_{\mathcal{T}}(\mathbf{s}, \hat{\mathbf{t}}) = \max_{q \in F} [\ C_q \cdot f(q)\ ]$

Coste: $\Theta(J \cdot |Q| \cdot \overline{|W_q|})$

En la práctica, sin embargo, el cálculo de $C(q)$ se restringe en cada etapa de modo que en vez de calcularlo para todo Q sólo se realiza para aquellos estados que son accesibles desde los que se han alcanzado previamente. Como el análisis comienza a partir de estados iniciales del modelo, este procedimiento garantiza que en cada etapa sólo se considera a aquellos estados alcanzables hasta dicha etapa, y por tanto con perspectivas de formar parte de un camino asociado a la cadena de entrada dada, pudiendo obviar el resto.

El algoritmo 2.1, tal y como se ha descrito, sólo proporciona el valor de $\max_{\mathbf{t} \in \Delta^\star} F_{\mathcal{T}}(\mathbf{s}, \mathbf{t})$, asociado al camino más probable de la cadena de entrada $\mathbf{s}$. Para conocer la derivación en sí misma, y así poder obtener el argumento $\hat{\mathbf{t}}$ que maximiza dicha expresión, es decir, la traducción más probable según $\mathcal{T}$, es necesario anotar qué alternativa fue escogida en cada decisión disponible, con el fin de poder conocer la secuencia de transiciones que se han utilizado. Este proceso incrementa la complejidad espacial proporcionalmente al número de etapas del *trellis*, lo que provoca que su cota sea de nuevo $\Theta(J \cdot |Q|)$.

Técnicas de poda

En aplicaciones prácticas, el coste temporal se puede reducir muy significativamente a través de la aplicación de la técnica conocida como búsqueda en haz, en la que sólo se consideran los caminos de mayor puntuación parcial.

A grandes rasgos, existen dos modos de podar la búsqueda, no incompatibles entre sí, que figuradamente controlan la amplitud del haz de exploración, es decir, la capacidad virtual de búsqueda a través de la red de estados finitos. Ambas aproximaciones pretenden acotar el número de vértices en cada etapa del *trellis*, evitando la exploración completa de Q, conjunto de estados de $\mathcal{T}$.

Así, según si el número de nodos por etapa se limita a un valor constante para todas las etapas o de si éste puede variar de una etapa a otra, se dice que la poda mediante búsqueda en haz es estática o dinámica, respectivamente. En la literatura, la poda estática se conoce como *histogram pruning* [STN94], mientras que una poda dinámica se suele citar como *beam search* [NMNP92].

Por un lado, la configuración de una búsqueda restringida mediante poda estática se reduce a instanciar el umbral h, con $1 \leq h \leq |Q|$, correspondiente al número de vértices que deseamos utilizar en cada etapa del *trellis*. De este modo, entre cada par de etapas consecutivas, las opciones de exploración que sobreviven de una a otra serán aquellas h con una mayor puntuación parcial.

Por otro lado, el modo de introducir restricciones dinámicas sobre el haz es a través de acotar las opciones de exploración en función de si su puntuación se encuentra en un determinado rango, generalmente determinado como un porcentaje de distancia respecto de la mejor opción de cada etapa. De esta manera, se evitan situaciones de discriminación en las que diversas opciones de exploración de puntuación muy similar puedan sufrir diferentes destinos (ser aceptadas o rechazadas) en el contexto de una búsqueda en haz.

Por supuesto, ambas aproximaciones son complementarias y en la práctica se emplean a la vez para limitar el consumo de recursos computacionales. Algorítmicamente, se aplica en primer lugar un haz dinámico sobre la exploración del modelo, restringiendo posteriormente su resultado por medio del

máximo número de vértices permitidos según el umbral de un haz estático. Lo deseable en este caso es que el haz estático no actúe salvo en casos extremos, permitiendo así la exploración completa sugerida por el haz dinámico.

2.2. Otros modelos de traducción

El problema de la traducción automática estadística se suele abordar en la literatura por medio de su descomposición según la regla de Bayes [BCDP+90]:

$$\Pr(\mathbf{t} \mid \mathbf{s}) = \frac{\Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t})}{\Pr(\mathbf{s})}$$

donde, de nuevo, el término del denominador $\Pr(\mathbf{s})$ es independiente del factor de maximización $\mathbf{t}$, por lo que el problema se puede resolver por medio de la combinación de sendas distribuciones de probabilidad, $\Pr(\mathbf{t})$ y $\Pr(\mathbf{s} \mid \mathbf{t})$:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}) = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \tag{2.9}$$

$\Pr(\mathbf{t})$ se aproxima a menudo mediante un modelo de lenguaje (véase la sección 3.1) para seleccionar aquellas frases mejor construidas en el idioma de salida y $\Pr(\mathbf{s} \mid \mathbf{t})$ se modela por medio de un modelo de traducción basado en diccionarios estocásticos y modelos de alineamientos [BPPM93, NNO+00].

Por otro lado, en la actualidad existen diversas aproximaciones basadas en un entorno de modelado log-lineal donde las distribuciones de probabilidad se modelan mediante una combinación de diversos modelos [ON03]:

$$\Pr(\mathbf{t} \mid \mathbf{s}) \approx \frac{\exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\mathbf{s}, \mathbf{t})\right)}{\sum_{\mathbf{t}'} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\mathbf{s}, \mathbf{t}')\right)}$$

y como el denominador tampoco depende de $\mathbf{t}$, se puede obviar en la búsqueda:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m \cdot h_m(\mathbf{s}, \mathbf{t}) \tag{2.10}$$

donde $h_m(\mathbf{s}, \mathbf{t})$ puede ser cualquier modelo que represente una característica importante en traducción, como por ejemplo $\log \Pr(\mathbf{s} \mid \mathbf{t})$, $\log \Pr(\mathbf{t} \mid \mathbf{s})$, $\log \Pr(\mathbf{t})$, etc., y M es el total de características empleadas en el modelado. Los parámetros λ_m son los pesos de los modelos en la combinación log-lineal.

Bajo este marco, los así llamados modelos basados en segmentos bilingües [OTN99, TC01, MW02, KOM03, KDB06] han demostrado proporcionar un entorno muy natural en traducción estadística. Calculando la probabilidad de traducción de un segmento (una secuencia de palabras), e introduciendo así información sobre el contexto dado, estos modelos han superado ampliamente a los modelos iniciales basados en palabras aisladas, convirtiéndose rápidamente en la tecnología predominante del estado del arte [KM06, Lop08]. Una variante de estos modelos son los llamados *alignment templates* [ON04]. Hay ciertos trabajos en los que, sin embargo, estos modelos se implementan como una aproximación basada en modelos de estados finitos [KB03, KB05]. Recientemente, este tipo de modelos se ha enriquecido mediante la incorporación de una cierta arquitectura jerárquica de corte sintáctico [Chi05, Chi07].

En las siguientes subsecciones, esta tecnología se describe más en detalle por medio de la formalización de los modelos pioneros, basados en palabras, su posterior extensión a segmentos, y los algoritmos de búsqueda propuestos.

2.2.1. Modelos de traducción basados en palabras

Los así llamados modelos de IBM, introducidos en [BPPM93], son una serie de 5 modelos de traducción sobre $\Pr(\mathbf{s} \mid \mathbf{t})$ de complejidad creciente, para una aproximación estadística a la traducción. Estos modelos se basan en el concepto de alineamiento entre las palabras de cada par de frases bilingües.

Sean J e I las longitudes respectivas en número de palabras de dos frases paralelas $\mathbf{s}$ y $\mathbf{t}$[2]. Formalmente, un alineamiento es una correspondencia entre

[2] Siguiendo la notación introducida en [BPPM93], un segmento de la forma $z_i, \ldots, z_j$ se denota mediante z_i^j, y para ciertos naturales N y M, la imagen de una función $f : \{1, ..., N\} \to \{1, ..., M\}$ para el número n se denota mediante f_n, y todos los posibles valores de la función f como f_1^N.

los conjuntos de posiciones de $\mathbf{s}_1^J$ y $\mathbf{t}_1^I$: $a \subset \{1 \cdots J\} \times \{1 \cdots I\}$. Sin embargo, en [BPPM93], el concepto de alineamiento se restringe para ser una función $a : \{1 \cdots J\} \rightarrow \{0 \cdots I\}$, en la que $a_j = 0$ significa que la posición j de $\mathbf{s}_1^J$ no se alinea con ninguna posición de $\mathbf{t}_1^I$ (o lo hace con la palabra nula $\mathbf{t}_0$). Todos los alineamientos posibles entre $\mathbf{s}_1^J$ y $\mathbf{t}_1^I$ se denotan mediante $\mathcal{A}(\mathbf{s}_1^J, \mathbf{t}_1^I)$ y su probabilidad de traducción a través de un alineamiento dado $a = a_1^J$ se denota mediante $\Pr(\mathbf{s}_1^J, a_1^J | \mathbf{t}_1^I)$. Por lo tanto, el término $\Pr(\mathbf{s} \mid \mathbf{t})$ de la ecuación (2.9) se puede obtener a partir de todos los posibles alineamientos entre $\mathbf{s}$ y $\mathbf{t}$:

$$\Pr(\mathbf{s} \mid \mathbf{t}) = \sum_{a \in \mathcal{A}(\mathbf{s},\mathbf{t})} \Pr(\mathbf{s}, a | \mathbf{t}) \tag{2.11}$$

A continuación, presentamos una breve descripción de las distribuciones de probabilidad implicadas en cada modelo de alineamiento a nivel de palabra descrito en [BPPM93]. Bajo un punto de vista generativo, $\Pr(\mathbf{s}, a | \mathbf{t})$ se puede descomponer por medio de la longitud de $\mathbf{s}$, J, de la siguiente manera:

$$\Pr(\mathbf{s}, a | \mathbf{t}) = \Pr(\mathbf{s}, J, a | \mathbf{t}) =$$

$$\Pr(J | \mathbf{t}) \cdot \Pr(\mathbf{s}, a | \mathbf{t}, J) =$$

$$\Pr(J | \mathbf{t}) \cdot \prod_{j=1}^{J} \Pr(a_j | \mathbf{s}_1^{j-1}, a_1^{j-1}, \mathbf{t}) \cdot \Pr(\mathbf{s}_j | \mathbf{s}_1^{j-1}, a_1^j, \mathbf{t}) \tag{2.12}$$

donde generalmente se asume para todos los modelos que la probabilidad de traducción $\Pr(\mathbf{s}_j \mid \mathbf{s}_1^{j-1}, a_1^j, \mathbf{t})$ se aproxima mediante $\Pr(\mathbf{s}_j \mid \mathbf{t}_{a_j})$, cuyo modelado se realiza habitualmente a través de una tabla conocida como diccionario estocástico. Las diferencias entre los modelos de alineamientos de IBM son con respecto a las asunciones tomadas sobre las probabilidades de alineamiento $\Pr(a_j | \mathbf{s}_1^{j-1}, a_1^{j-1}, \mathbf{t})$, las cuales se pueden resumir del siguiente modo:

- Para el modelo 1, la distribución es uniforme.
- Para el modelo 2, se aproxima mediante $\alpha(a_j | j, J, I)$, un modelo de alineamiento de orden 0 que establece una dependencia absoluta entre las posiciones de las palabras de entrada y las de las palabras de salida.

- Para los modelos 3, 4 y 5, se introduce un modelo de fertilidad $f(\phi|\,\mathbf{t}_i)$ que denota la probabilidad de que la palabra $\mathbf{t}_i$ genere un determinado número ϕ de palabras. La probabilidad de alineamiento del modelo 3 se aproxima mediante un modelo de distorsión de orden 0 $\delta(j|a_j, J, I)$, en el que también se establecen dependencias absolutas entre las posiciones de las palabras. En cambio, los modelos 4 y 5 utilizan dos distribuciones para representar un modelo de distorsión de primer orden, lo que posibilita, entre otras cosas, el empleo de dependencias relativas.

Una de las aplicaciones más interesantes de esta aproximación consiste en la obtención del alineamiento de mayor probabilidad $\hat{a}$ que relaciona $\mathbf{s}$ y $\mathbf{t}$, cuya estimación se puede calcular a partir del aprendizaje de estos modelos.

Todo lo necesario para estimar los modelos de alineamientos de IBM se describe en [BPPM93]. La herramienta pública GIZA++ [ON03] es una posible implementación de dicha estimación, donde a través del algoritmo *Expectation-Maximization* [DLR77], los modelos 1 y 2 se aprenden de forma exacta, mientras los modelos 3, 4, y 5 se infieren mediante una aproximación.

2.2.2. Modelos de traducción basados en segmentos

La derivación de los modelos de traducción basados en segmentos proviene del concepto de segmentación bilingüe. Un segmento se define como una secuencia de palabras consecutivas que suceden en el contexto de una frase. Por lo tanto, resulta igualmente apropiado decir que una frase es tanto una secuencia de palabras como una secuencia de segmentos, en la que cada uno de ellos está compuesto a su vez de un cierto número de palabras ordenadas.

Dado un par de frases paralelas, una segmentación bilingüe es una representación de dichas frases a través de un mismo número de segmentos K, de manera que cada segmento definido ha de encontrarse biunívocamente relacionado con su segmento correspondiente en el otro idioma, y viceversa.

Si añadimos una restricción de monotonicidad a la definición de segmentación bilingüe, es decir, impidiendo aquellas relaciones entre segmentos

fuente y destino que se cruzaran gráficamente entre sí, entonces podremos trasladar el formalismo de los modelos de traducción basados en segmentos hacia un marco de trabajo basado en modelos de estados finitos, debido a que los transductores de estados finitos son modelos de traducción monótonos.

El concepto de segmentación bilingüe se formaliza a través de dos funciones de segmentación, μ y ρ, una para cada una de las frases que segmentar:

$$\mu_0^K \rightarrow \{0,\ldots,J\} \qquad \rho_0^K \rightarrow \{0,\ldots,I\}$$

$$\mu_0 = 0 \quad \mu_K = J \qquad \rho_0 = 0 \quad \rho_K = I$$

$$\forall k = 1 \ldots K : \mu_k \geq \mu_{k-1} \ \wedge \ \rho_k \geq \rho_{k-1}$$

Ya que sólo nos interesan las segmentaciones de carácter monótono, se puede establecer que, para todo $k = 1 \ldots K$, los segmentos $\mathbf{s}_{\mu_{k-1}+1}^{\mu_k}$ y $\mathbf{t}_{\rho_{k-1}+1}^{\rho_k}$ deben estar biunívocamente relacionados entre sí.

Por lo tanto, y asumiendo que las distribuciones de probabilidad que gobiernan las segmentaciones son uniformes, de modo que todas las segmentaciones de ambas frases en K posibles segmentos son equiprobables, y a la vez independientes de K, entonces el término $\Pr(\mathbf{s} \mid \mathbf{t})$ se puede reescribir como:

$$\Pr(\mathbf{s} \mid \mathbf{t}) \propto \sum_K \sum_{\mu_0^K} \sum_{\rho_0^K} \prod_{k=1}^{K} \Pr(\mathbf{s}_{\mu_{k-1}+1}^{\mu_k} | \mathbf{t}_{\rho_{k-1}+1}^{\rho_k}) \tag{2.13}$$

Aprendizaje de modelos de traducción basados en segmentos

En última instancia, un modelo de traducción basado en segmentos no es más que un diccionario estocástico con una granularidad a nivel de segmento. Es decir, es una tabla de tres columnas cuyas entradas respetan este patrón:

$$\mathbf{s}_j \ldots \mathbf{s}_{j'} ||| \mathbf{t}_i \ldots \mathbf{t}_{i'} ||| \Pr(\mathbf{s}_j \ldots \mathbf{s}_{j'} | \mathbf{t}_i \ldots \mathbf{t}_{i'})$$

en el que la primera columna representa el segmento de entrada, la segunda, el segmento de salida, y la tercera expresa la probabilidad asignada a dicho par de segmentos.

En los últimos años, se ha investigado e implementado una amplia variedad de técnicas para producir diccionarios basados en segmentos [KOM03]. En primer lugar, en [TC01, MW02, Tom03] se propuso una estimación directa de los parámetros de $\Pr(\mathbf{s}_j^{j'} \mid \mathbf{t}_i^{i'})$ siguiendo el criterio de máxima verosimilitud, con ayuda de un corpus paralelo alineado a nivel de frase. Por otro lado, en [ZON02, OGVC05, KDB06] se utiliza una serie de heurísticos para extraer todas las segmentaciones bilingües que sean consistentes con un corpus paralelo alineado a nivel de palabra mediante la herramienta GIZA++. Finalmente, una parte importante de la comunidad investigadora ha sugerido que el procedimiento de extracción de segmentos se realice siguiendo ciertas motivaciones lingüísticas, por lo que sólo se tendrían en cuenta aquellos segmentos que cumplieran una serie de reglas lingüísticas determinadas, a los que se conoce ampliamente en la literatura de investigación como *chunks*.

En esta tesis, el entrenamiento de modelos de traducción basados en segmentos se ha realizado íntegramente a través de diversos algoritmos de extracción de segmentos, de naturaleza heurística, y a partir de ciertos alineamientos a nivel de palabra obtenidos estadísticamente por medio de GIZA++.

Hoy en día, sin embargo, el uso de modelos basados en segmentos se está integrando en arquitecturas de traducción más complejas como las aproximaciones basadas en modelos factoriales [KH07] o jerárquicos [Chi05, Chi07].

2.2.3. Descodificación usando modelos de traducción

Dada una frase de entrada $\mathbf{s}_1^J$, el objetivo de la fase de búsqueda en traducción automática estadística es encontrar una frase de salida $\hat{\mathbf{t}}_1^{\hat{I}}$ que maximice el producto $\Pr(\hat{\mathbf{t}}_1^{\hat{I}}) \cdot \Pr(\mathbf{s}_1^J \mid \hat{\mathbf{t}}_1^{\hat{I}})$. A lo largo de la literatura se han propuesto diferentes algoritmos de búsqueda. La idea básica de la mayoría de ellos es la de generar hipótesis parciales de un modo incremental. Cada hipótesis representa un prefijo de $\hat{\mathbf{t}}_1^{\hat{I}}$, al que se le asocia un subconjunto de posiciones de entrada con las que se alinea, junto con una puntuación parcial. A partir de la extensión del prefijo correspondiente de una hipótesis, se pueden generar

nuevas hipótesis, con una cobertura cada vez mayor sobre la frase de entrada.

Para definir la organización del espacio de búsqueda, se han sugerido diversas estrategias de exploración. Algunos autores [OGVC03, GJK$^+$01] han propuesto el uso de un algorimo $A^\star$ para organizar el espacio de búsqueda, en el que utilizando una pila de prioridad se adopta una estrategia primero-el-mejor. Por otro lado, en [BBDP$^+$96] se sugiere una estrategia primero-en-profundidad mediante el uso de una serie de pilas. Concretamente, el algoritmo emplea una pila diferente para almacenar las hipótesis en función del subconjunto de posiciones de entrada que se han utilizado para generar el prefijo correspondiente. Este procedimiento permite forzar la expansión de hipótesis con diferente grado de completitud. En cada iteración, el algoritmo explorará todas las pilas extendiendo la mejor hipótesis de cada una de ellas.

Asimismo, en [GJK$^+$01, Ger03] también se propuso un algoritmo de búsqueda *greedy*. Su principal diferencia reside en que no sigue un proceso incremental para construir la hipótesis de salida. En su lugar, el algoritmo comienza con una hipótesis inicial completa, que se construye mediante la mejor traducción individual de las palabras de entrada. Iterativamente, se aplica una serie de operaciones sobre las hipótesis para obtener unas nuevas candidatas.

Otros autores sugieren una aproximación mediante programación dinámica [TN03] en la que la idea clave es que el proceso de búsqueda se puede sincronizar con el número de posiciones de entrada que ya se han traducido.

2.3. Resumen del capítulo

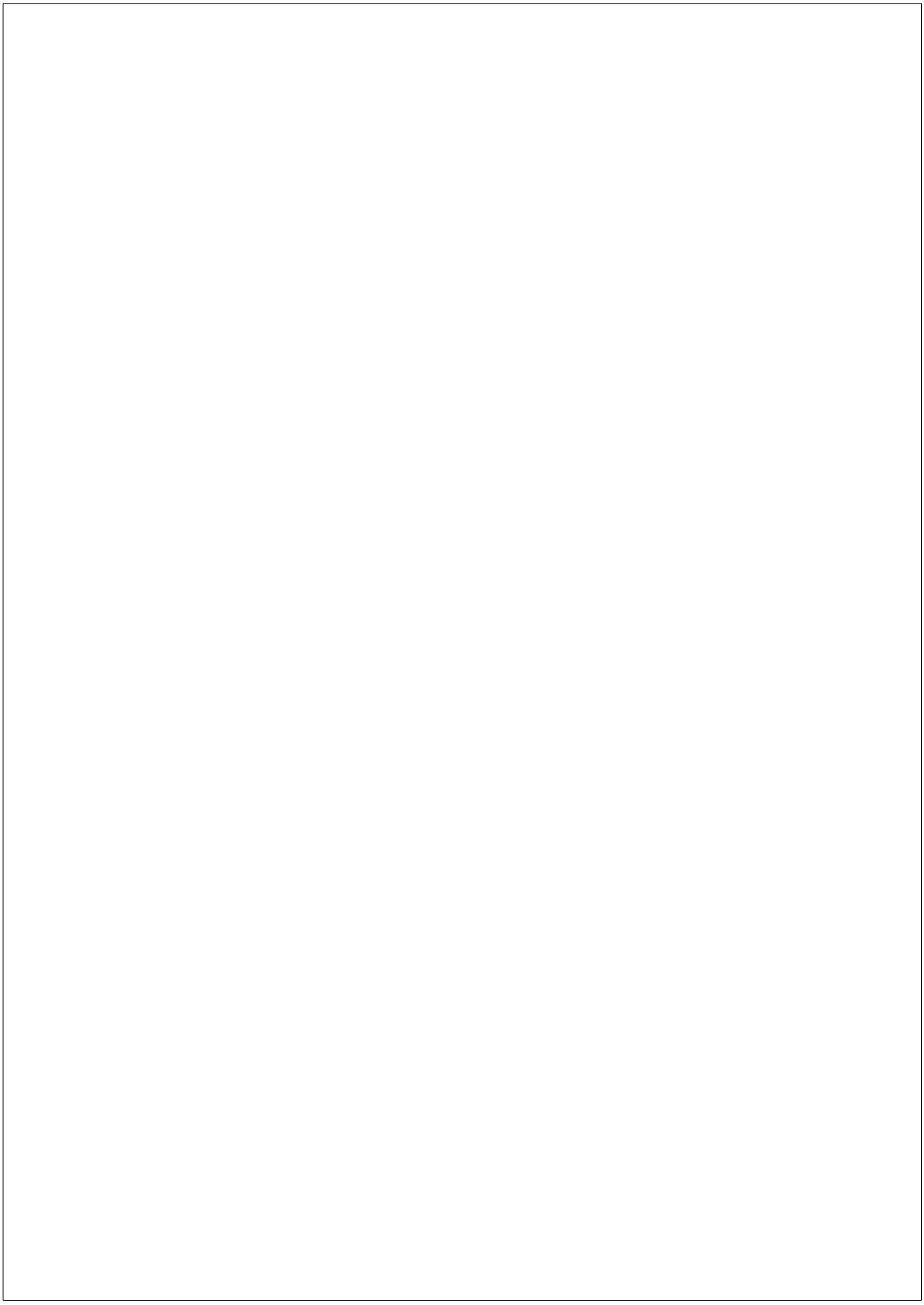
Este capítulo ha introducido formalmente los modelos estadísticos en los que se basa esta tesis para su aplicación en el área de Traducción Automática:

- Por un lado, los autómatas de estados finitos y su extensión estocástica se han definido como formalismo de expresión de lenguajes regulares.
- A su vez, los transductores de estados finitos y su extensión estocástica se han definido como formalismo de expresión de relaciones regulares.

> El transductor es el modelo básico de traducción sobre el que se desarrolla esta tesis. Sin embargo, la definición de éstos como una extensión a partir de los autómatas no sólo obedece a términos expresivos sino también a la proposición que relaciona ambos modelos y que da lugar a la metodología de aprendizaje que se expone en el siguiente capítulo.

Además, se ha descrito en detalle el procedimiento de búsqueda mediante modelos de estados finitos, junto con el uso de sus técnicas de poda más habituales. Estos algoritmos tienen la particularidad de que son aplicables indistintamente sobre los dos tipos de modelos presentados, es decir, que sirven tanto para realizar la búsqueda a través de autómatas como de transductores.

Por último, este capítulo ha expuesto asimismo la aproximación bayesiana a la traducción, debido sobre todo a que diversos elementos de la misma, como los alineamientos estadísticos o los modelos basados en segmentos, forman parte integrante de nuestros métodos de inferencia de transductores.



Capítulo 3

GIATI como metodología de aprendizaje

La metodología GIATI [Cas00, CVP05] se ha revelado como una aproximación interesante para la inferencia de transductores estocásticos de estados finitos a través del modelado de lenguajes. En lugar de aprender relaciones entre las palabras de entrada y salida, GIATI convierte en primer lugar cada par de frases paralelas en una única cadena de símbolos extendidos para, acto seguido, inferir un modelo de lenguaje a partir del nuevo corpus creado. De hecho, las ideas de GIATI, que se introdujeron para su aplicación sobre modelos de estados finitos, son realmente extensibles a cualquier otra técnica de modelado del lenguaje. Como ejemplo, la filosofía GIATI se ha desarrollado bajo un marco de trabajo basado en combinaciones lineales de modelos en el que se emplea un modelo de lenguaje de estas características [MBC$^+$06]. Sin embargo, volviendo a nuestra aproximación, el algoritmo GIATI se implementará mediante autómatas y transductores estocásticos de estados finitos.

A continuación, introducimos una sección sobre modelado del lenguaje, donde describimos una de sus aproximaciones más populares, los n-gramas. Ha sido la opción elegida para implementar en la práctica esta metodología a través de su representación equivalente usando modelos de estados finitos.

3.1. Modelos de lenguaje

Los modelos de lenguaje son un intento de capturar las regularidades existentes en lenguaje natural utilizando distribuciones de probabilidad para reflejar ciertos eventos lingüísticos tales como la aparición de una palabra en un contexto dado, bien sea éste una frase, un párrafo, o incluso un artículo. Los modelos de lenguaje se utilizan ampliamente en diversas aplicaciones de las tecnologías del lenguaje en las que su participación ha demostrado ser de vital importancia. Por ejemplo, en reconocimiento del habla, traducción automática, reconocimiento de texto manuscrito, clasificación de documentos, recuperación de información, corrección ortográfica, etc. Un estudio de carácter prospectivo sobre modelos de lenguaje se puede encontrar en [Ros00, Goo03].

Un modelo de lenguaje estima la distribución de probabilidad $\Pr(\overline{\gamma})$ sobre las cadenas que pertenecen a un cierto lenguaje. El alfabeto de estos modelos suele representar a algún conjunto de elementos de naturaleza lingüística, como por ejemplo, palabras, caracteres, fonemas, o categorías morfosintácticas.

3.1.1. n-gramas

En la actualidad, los modelos de lenguaje más utilizados en la práctica son los así llamados modelos de n-gramas [Jel98], en los que la formación de cadenas se modela siguiendo un proceso estocástico de Markov de orden $n-1$:

$$\begin{aligned}
\Pr(\overline{\gamma}) &= \prod_{i=1}^{|\overline{\gamma}|} \Pr(\gamma_i|\gamma_1 \ldots \gamma_{i-1}) \\
&= \Pr(\gamma_1)\Pr(\gamma_2|\gamma_1)\Pr(\gamma_3|\gamma_1\gamma_2)\ldots\Pr(\gamma_{|\overline{\gamma}|}|\gamma_1\gamma_2\ldots\gamma_{|\overline{\gamma}|-1}) \\
&\approx \prod_{i=1}^{|\overline{\gamma}|} \Pr(\gamma_i|\gamma_{i-n+1}\ldots\gamma_{i-1}) \qquad (3.1)
\end{aligned}$$

limitando la influencia de la historia previa a los $n-1$ símbolos anteriores.

Un modelo de n-gramas es un caso particular de autómata estocástico, aunque, por comodidad, la notación empleada es la habitual en la literatura, y no la introducida en el capítulo anterior sobre autómatas de estados finitos.

La estimación de los parámetros del modelo se reduce por tanto al aprendizaje de las distribuciones de probabilidad $\Pr(\gamma_i|\gamma_{i-n+1}\dots\gamma_{i-1})$, asociadas a las secuencias correspondientes de n símbolos consecutivos $\gamma_{i-n+1}\dots\gamma_{i-1}\gamma_i$, conocidas con el nombre de n-gramas. Generalmente, un modelo de n-gramas se puede inferir mediante un corpus de texto utilizando técnicas de máxima verosimilitud. Simplificando bastante el proceso, básicamente se compone de un conteo del número de ocurrencias de los distintos n-gramas del texto, normalizado adecuadamente mediante la suma de las frecuencias de todos los n-gramas que comparten su misma historia, difiriendo en su último símbolo:

$$\Pr(\gamma_i|\gamma_{i-n+1}\dots\gamma_{i-1}) = \frac{c(\gamma_{i-n+1}\dots\gamma_{i-1}\gamma_i)}{\sum_{\gamma_{i'}} c(\gamma_{i-n+1}\dots\gamma_{i-1}\gamma_{i'})}$$

siendo $c(x)$ el número de veces que el n-grama x sucede a lo largo del texto.

Sin embargo, esta estimación sufre de un problema de dispersión de datos. Según [Ros96], tras observar todos los trigramas de un corpus periodístico de 38 millones de palabras, contrastándolos con los de otro corpus de la misma naturaleza y origen, un tercio de estos últimos no suceden en el primero. Generalmente, las distribuciones de probabilidad se deben suavizar de modo que se les permita contemplar los elementos no vistos en el entrenamiento, con el objetivo de proporcionar una cobertura máxima sobre los datos de entrada, es decir, para poder otorgar una probabilidad a cualquier cadena $\overline{\gamma}$.

Uno de los métodos más comunes de suavizado consiste en asignar cierta probabilidad a los eventos desconocidos, descontada previamente de los sucesos observados, combinado con una jerarquía de pesos de *backoff* para ponderar el cálculo de la probabilidad de un símbolo $\Pr(\gamma_i|\gamma_{i-n+1}\dots\gamma_{i-1})$ en función del n-grama de mayor orden que esté contemplado en el modelo. Por ejemplo, sea $\mathcal{M}$ un modelo de trigramas suavizado mediante *backoff* por medio de los modelos de bigramas y unigramas correspondientes, el cálculo de la probabilidad de cierto trigrama abc, $\Pr(c|ab)$, se obtendría de este modo:

$$\Pr(c|ab) = \begin{cases} \text{si} & abc \in \mathcal{M} & : & \Pr_{\mathcal{M}}(c|ab) \\ \text{sino si} & bc \in \mathcal{M} & : & Bo_{\mathcal{M}}(ab) \cdot \Pr_{\mathcal{M}}(c|b) \\ \text{sino si} & c \in \mathcal{M} & : & Bo_{\mathcal{M}}(ab) \cdot Bo_{\mathcal{M}}(b) \cdot \Pr_{\mathcal{M}}(c) \\ \text{sino} & & : & Bo_{\mathcal{M}}(ab) \cdot Bo_{\mathcal{M}}(b) \cdot \Pr_{\mathcal{M}}(unk) \end{cases}$$

donde $\Pr_{\mathcal{M}}$ es la probabilidad estimada por el modelo para el n-grama correspondiente, $Bo_{\mathcal{M}}$ es el peso de *backoff* que el modelo emplea para el suavizado de los eventos no vistos durante el entrenamiento, y, por último, $\Pr_{\mathcal{M}}(unk)$ es la masa de probabilidad reservada para símbolos desconocidos.

La utilización de un suavizado mediante *backoff* en modelos de n-gramas para recurrir a modelos de orden inferior se emplea en [Kat87, NEK94, KN95]. Otros métodos utilizan interpolación lineal [JM80], n-gramas de longitud variable [Kne96], redes finitas [DR97], o técnicas de máxima entropía [BDPDP96].

3.2. Inferencia de transductores estocásticos

GIATI se apoya para su enunciación en un conocido teorema que relaciona los lenguajes locales y los regulares a través de homomorfismos. Así, los lenguajes regulares que induce un transductor de estados finitos por medio de sus alfabetos de entrada Σ y de salida Δ se pueden expresar mediante un nuevo alfabeto común Γ a través de los morfismos correspondientes [CVP05].

Más concretamente, dado un corpus paralelo consistente en una muestra finita Z de pares de cadenas: primeramente, cada par de entrenamiento $(\mathbf{s}, \mathbf{t}) \in \Sigma^\star \times \Delta^\star$ se transforma en una cadena $\overline{\gamma} \in \Gamma^\star$ de un alfabeto extendido, produciendo un corpus S; en segundo lugar, se infiere un autómata estocástico de estados finitos $\mathcal{A}$ a partir de S; por último, se revierte la transformación inicial sobre los símbolos de Γ, restaurando los pares de cadenas de entrada y de salida pertenecientes a $\Sigma^\star \times \Delta^\star$, convirtiendo, por lo tanto, al autómata estocástico $\mathcal{A}$ en un transductor estocástico de estados finitos $\mathcal{T}$. Otros autores, no obstante, utilizan esta metodología con transductores que no exigen que sus distribuciones de probabilidad sean consistentes [KVM$^+$05].

La primera transformación se modela por medio de cierta función de etiquetado $\mathcal{L} : \Sigma^\star \times \Delta^\star \rightarrow \Gamma^\star$, mientras que la última conversión se define a través de una función de etiquetado inverso $\Lambda(\cdot)$, de manera que $\Lambda(\mathcal{L}(Z)) = Z$. La construcción de un corpus de símbolos extendidos a partir del corpus bilingüe original permite el uso de múltiples algoritmos para el aprendizaje de autómatas estocásticos de estados finitos (o de modelos equivalentes) que han sido propuestos en la literatura de investigación sobre inferencia gramatical.

Las distintas funciones de etiquetado utilizadas en el contexto de esta tesis se basan en el concepto de segmentación monótona entre frases paralelas, también conocida como segmentación bilingüe monótonamente restringida, cuya definición se introdujo formalmente en el transcurso de la sección 2.2.2 mediante dos funciones que dividen el par $(\mathbf{s}, \mathbf{t})$ en un número K de trozos. Por lo tanto, cada par de segmentos $(\bar{s}_k, \bar{t}_k)$ es un símbolo del alfabeto de Γ, y la cadena de símbolos extendidos $\overline{\gamma} \in \Gamma^\star$ se construye del siguiente modo:

$$K > 0$$

$$\mathbf{s} = \bar{s}_1 \ldots \bar{s}_K$$
$$\mathbf{t} = \bar{t}_1 \ldots \bar{t}_K$$

$$\overline{\gamma} = \gamma_1 \cdots \gamma_K$$

de manera que $\forall k = 1 \ldots K \ : \ \gamma_k = (\bar{s}_k, \bar{t}_k) \in \Sigma^\star \times \Delta^\star$.

Ilustremos con un sencillo ejemplo el funcionamiento de esta metodología. Dadas diversas muestras de una tarea de traducción entre español e inglés:

El coche rojo $\Leftrightarrow$ The red car

El coche $\Leftrightarrow$ The car

El coche azul $\Leftrightarrow$ The blue car

asumamos un proceso de segmentación cuyo resultado haya sido el siguiente:

	$K=3$			$K=2$		$K=3$		
	1	2	3	1	2	1	2	3
s	El	coche	rojo	El	coche	El	coche	azul
t	The	ε	red car	The	car	The	ε	blue car

luego las cadenas $\overline{\gamma}$ se construyen así: $\overline{\gamma}_1 =$ (El, The) (coche, ε) (rojo, red car), $\overline{\gamma}_2 =$ (El, The) (coche, car), y $\overline{\gamma}_3 =$ (El, The) (coche, ε) (azul, blue car). Nótese, por ejemplo, que las frases originales se pueden reconstruir por medio de la concatenación de las cadenas que existen entre los símbolos extendidos de $\overline{\gamma}$.

A partir de aquí, cualquier método de inferencia de autómatas estocásticos produciría un modelo cuyas etiquetas de transición tendrían el formato $(\overline{s}, \overline{t})$, que permiten identificar la entrada y salida asociada al transductor derivado. El modelo de lenguaje inferido a partir del corpus de símbolos extendidos estima las probabilidades de concatenación de tales etiquetas compuestas. En la figura 3.1 se puede observar la composición de un modelo de este tipo.

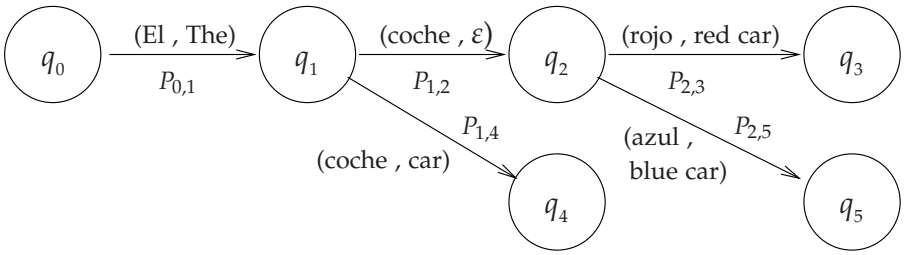


Figura 3.1: Detalle del autómata inferido a partir de las cadenas extendidas. La derivación del transductor correspondiente es un procedimiento trivial dado que cada etiqueta se define como un par de cadenas de entrada y salida.

En definitiva, uno de los factores más determinantes sobre el rendimiento de GIATI como constructor de modelos de traducción es la definición de Γ. Cada símbolo del alfabeto extendido debe condensar de algún modo la relación significativa que existe entre las palabras de entrada y salida. El descubrimiento de estas relaciones es un problema que se ha estudiado a fondo en el área de traducción automática estadística, dando lugar a una serie de técnicas bien establecidas para su tratamiento. El concepto de alineamiento estadístico formaliza este problema. Si éste se restringe a una correspondencia uno-a-uno, uno-a-muchos, o muchos-a-muchos, dependerá de las asunciones particulares que se deseen realizar. La restricción de la función de alineamien-

to simplifica el procedimiento de aprendizaje pero provoca una pérdida en el poder de expresión del modelo. Los algoritmos existentes tratan de encontrar un compromiso eficiente entre la complejidad del modelo y su expresividad.

3.2.1. Instanciación de GIATI mediante modelos de n-gramas

Nuestra implementación de GIATI utiliza un modelo de n-gramas suavizado mediante *backoff*, cuya expresión en términos de un autómata estocástico de estados finitos es completamente equivalente [Llo00]. Como modelo alternativo a los n-gramas, otros autores [PGCT04, PCTG05, PTCG06] usan GIATI mediante gramáticas k-explorables en sentido estricto. La figura 3.2 muestra un esquema global para la representación de modelos de n-gramas suavizados mediante *backoff* a través de modelos estructurales de estados finitos.

Dicha estructura, que es una representación compacta pero sin pérdidas del autómata estocástico de estados finitos que equivale a una jerarquía de modelos de n-gramas suavizados mediante *backoff*, posee una serie de características particulares. Por un lado, cada estado está asociado a una cadena de símbolos que representa una posible historia del modelo de n-gramas. Conceptualmente, los estados pueden quedar de este modo agrupados en niveles de altura en función de la longitud de su cadena de historia asociada. El nivel inferior contiene un único estado, asociado a la cadena vacía ε, designando a la historia que comparten todas las probabilidades unigrama. Por tanto, la capa más alta se sitúa en el nivel $(n-1)$ y contiene aquellos estados asociados a las historias de las probabilidades n-grama correspondientes. El estado inicial del modelo, denotado mediante <s>, se ubica en el nivel 1, donde también están el estado <unk>, asociado a los eventos desconocidos, y, por definición del modelo (y denotado como </s>), su único estado final.

Por otro lado, las transiciones entre estados representan la actualización de la historia ante una ocurrencia del símbolo de transición correspondiente. Dicha actualización se define de forma inequívoca mediante la concatenación entre la cadena correspondiente a la historia asociada al estado de origen y el símbolo de la etiqueta de transición, limitando el resultado a la subcadena

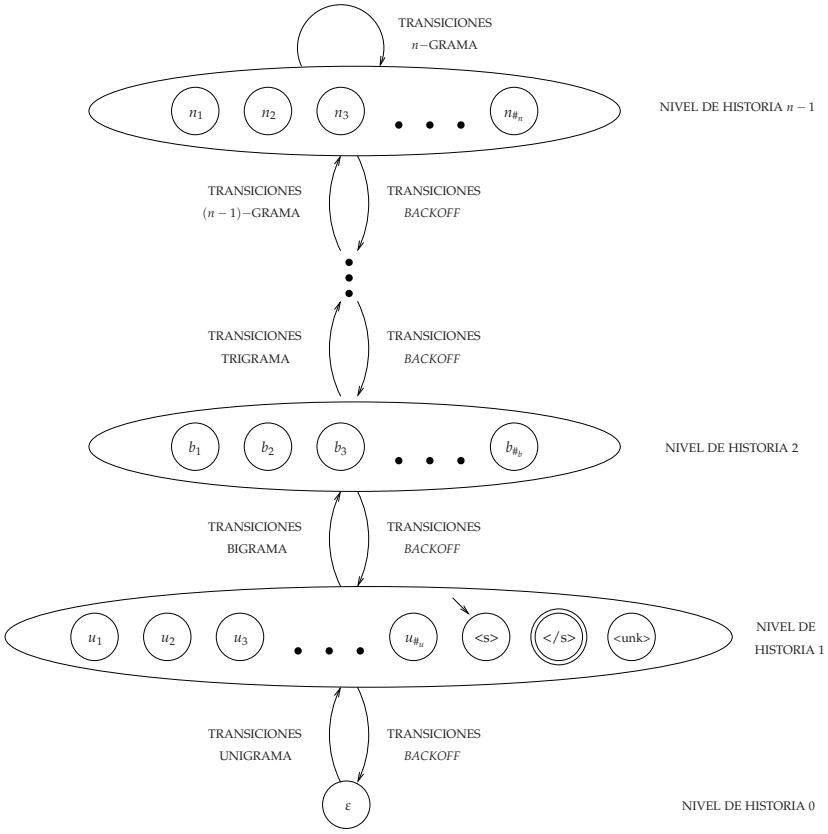


Figura 3.2: Un modelo de n-gramas con *backoff* en forma de autómata

de la derecha más larga cuya longitud sea, como máximo, de $n-1$ símbolos. Por tanto, la red de estados finitos de la figura 3.2 es un modelo determinista, y como tal, es no ambiguo, y sus cadenas tienen un único camino asociado.

Por este motivo, la probabilidad $\Pr(\overline{\gamma})$ se puede calcular de forma exacta por medio del algoritmo de Viterbi, donde la algorítmica definida para $\Pr(\gamma_i|\gamma_{i-n+1}\dots\gamma_{i-1})$ se implementa mediante una serie de aristas de fallo, con ε como etiqueta y el peso de *backoff* correspondiente como probabilidad. Las transiciones de *backoff* representan la pérdida del símbolo más lejano de

la historia ante la ocurrencia de una situación que no se recoge en el modelo. Por lo tanto, dichas aristas unen los estados correspondientes a sus historias.

El modelo estocástico asumido por este autómata es de tipo generativo, ya que la probabilidad que se modela en cada estado q es $\Pr(\gamma_i|q)$, lo que identifica unívocamente al estado de destino q', es decir, $\Pr(\gamma_i|q) = \Pr(q', \gamma_i|q)$, que es la distribución asociada a modelos generativos (véase la sección 2.1.2).

Al separar las etiquetas del autómata, obteniendo el transductor asociado, algunas de estas propiedades se mantienen y otras, en cambio, desaparecen. Obviamente, el determinismo y la no ambigüedad en el transductor derivado se conservan respecto de símbolos y cadenas de entrada y salida a la vez, ya que la red estructural del transductor es la misma que la del autómata. Sin embargo, estas propiedades se pierden sobre los alfabetos por separado. Estocásticamente, el transductor inferido continúa siendo de tipo generativo.

La conversión de estos transductores hacia un proceso estocástico aceptor, mediante la normalización de las probabilidades de transición pertinentes, se ha evaluado experimentalmente con resultados negativos. Por esta razón, dicha aproximación se ha abandonado por completo, dirigiendo la atención a esta implementación, que construye modelos generativos de forma natural.

Prosigamos la ilustración de GIATI mediante la utilización de bigramas como modelo de lenguaje sobre el ejemplo que se introdujo en la sección 3.2. El alfabeto Γ para esta tarea podría incluir entre sus símbolos a los siguientes:

$$\begin{aligned} \gamma_0 &= \text{<s>} \\ \gamma_1 &= \text{(El, The)} \\ \gamma_2 &= \text{(coche, } \varepsilon\text{)} \\ \gamma_3 &= \text{(rojo, red car)} \\ &\dots \end{aligned}$$

de manera que $\overline{\gamma}_1 = \gamma_1\gamma_2\gamma_3$ es una posible transformación del primer par de frases del ejemplo propuesto en una única cadena de símbolos extendidos. Así, a partir de la observación de la secuencia de símbolos que suceden en $\overline{\gamma}_1$, un modelo suavizado de bigramas tendría en cuenta los siguientes eventos:

unigramas			bigramas		
P_1	=	$\Pr(\gamma_1)$	$P_{0,1}$	=	$\Pr(\gamma_1\|\gamma_0)$
P_2	=	$\Pr(\gamma_2)$	$P_{1,2}$	=	$\Pr(\gamma_2\|\gamma_1)$
P_3	=	$\Pr(\gamma_3)$	$P_{2,3}$	=	$\Pr(\gamma_3\|\gamma_2)$

cuya expresión como un transductor estocástico se puede ver en la figura 3.3.

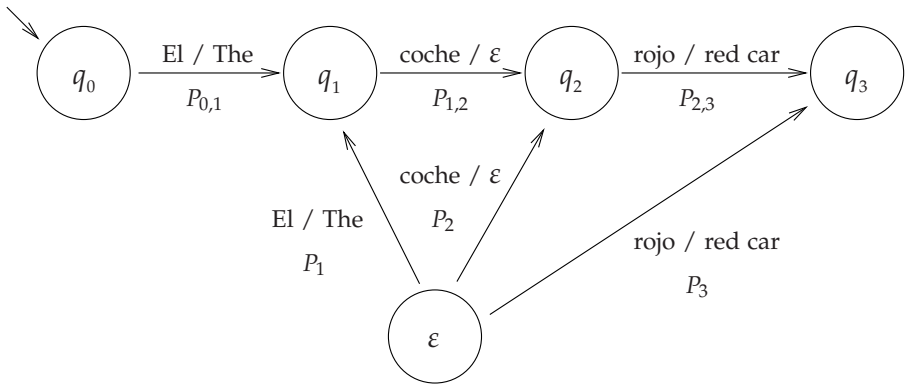


Figura 3.3: Transductor generado desde un modelo suavizado de bigramas. Para una mejor percepción, los arcos de *backoff* no se constatan en el modelo. Dichas aristas permiten el acceso desde el nivel superior al inferior transitando desde cualquier estado q_i al estado ε mediante una cierta probabilidad. Los estados del nivel superior se asocian a cadenas de historia de 1 símbolo, es decir, a la etiqueta de sus aristas entrantes, que es la misma para todas. Por lo tanto, $q_0 =$ <s>, $q_1 =$ (El, The), $q_2 =$ (coche, ε), y $q_3 =$ (rojo, red car).

A continuación, explicamos un método para el filtrado de transductores cuyo objetivo es reducir el transductor derivado de un modelo de n-gramas al ámbito de actuación sobre una determinada partición de datos a traducir. Esta técnica de poda está basada en una detección de los n-gramas bilingües cuya cadena de entrada asociada no está representada en el conjunto de test, lo que permite conocer a priori las estructuras inalcanzables mediante dicho corpus, y así descartarlas del modelo.

Poda de n-gramas para la reducción de transductores

Este método se inspira en otros sistemas de traducción automática [Koe04, KHB+07], los cuales habitualmente filtran sus modelos de traducción por medio de las frases del conjunto de test.

Dado que un evento n-grama $\gamma_{i-n+1}\gamma_{i-n+2}\ldots\gamma_{i-1}\gamma_i$ se expresa estadísticamente como $\Pr(\gamma_i|\gamma_{i-n+1}\gamma_{i-n+2}\ldots\gamma_{i-1})$, dicho evento se representará en estados finitos mediante una transición entre los estados correspondientes a la historia anterior y posterior a la aparición del símbolo γ_i en dicho contexto. La figura 3.4 refleja la representación de n-gramas como aristas entre estados.

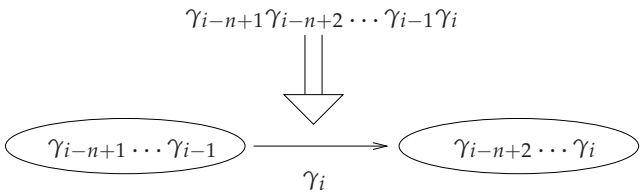


Figura 3.4: Representación de eventos n-grama mediante estados finitos

Por lo tanto, para poder utilizar dicha transición, la búsqueda debe haber alcanzado el estado $\gamma_{i-n+1}\ldots\gamma_{i-1}$ y la frase de entrada pendiente de análisis debe ser compatible con los símbolos de entrada de γ_i. En otras palabras, la secuencia completa de entrada correspondiente al evento n-grama $\gamma_{i-n+1}\ldots\gamma_{i-1}\gamma_i$ debe estar presente en el conjunto de test. Si no, es imposible que dicha transición pueda usarse durante el análisis de dicho corpus. Como resultado, todos los eventos n-grama cuya secuencia de entrada no se encuentre en el conjunto de test pueden obviar el proceso de generación de la transición correspondiente, ya que la ausencia de dichas aristas no afecta al proceso de descodificación, reduciendo, por tanto, el tamaño de los modelos. En el caso de eventos n-grama cuyo orden n no sea el máximo del modelo, la situación es similar, e incluso se puede obviar también la construcción de la arista asociada a su probabilidad de *backoff*, ya que dicha transición partiría de un estado que no es alcanzable por medio del susodicho conjunto de test.

Sin embargo, ya que el número de palabras de entrada de un n-grama de símbolos extendidos no se encuentra acotado, la verificación de su presencia o ausencia dentro del conjunto de test implicaría el almacenamiento de todas las subcadenas de cada frase, lo que no resulta viable en la práctica. En su lugar, se utiliza una ventana de tamaño variable para almacenar las palabras del conjunto de test, y cada evento n-grama se examina por medio de la aplicación de dicha ventana sobre la proyección monolingüe de entrada correspondiente, detectando si la generación de aristas puede obviarse o no. La descripción de esta técnica de poda se puede observar en el algoritmo 3.1.

Algoritmo 3.1 Poda de n-gramas para la reducción de transductores

Datos
  Z es un corpus con un total de X palabras de entrada
  V es el tamaño de ventana empleado
  $\mathcal{M}$ es un modelo de n-gramas sobre el alfabeto Γ
Resultado $\mathcal{T}$
Método
  $\forall x = 1 \dots X$
    $\forall v = 1 \dots V$
      $hash\_insertar(Z_{x-v+1} \dots Z_x)$
  $inicializar(\mathcal{T})$
  $\forall n' = 1 \dots n$
    $\forall \overline{\gamma} = \gamma_{i-n'+1} \dots \gamma_i \in \mathcal{M}_{n'}$ `(a)`
      $\mathbf{s}_1^J = proyectar\_entrada(\overline{\gamma})$
      $\forall j = 1 \dots J$
        $\forall v = 1 \dots V$
          si $(hash\_buscar(\mathbf{s}_{j-v+1} \dots \mathbf{s}_j) = \text{FALSO})$
            volver a `(a)`
      $\mathcal{T} = procesar(\overline{\gamma}, \Pr_{\mathcal{M}}(\gamma_i | \gamma_{i-n'+1} \dots \gamma_{i-1}), Bo_{\mathcal{M}}(\overline{\gamma}), \mathcal{T})$
Coste: $\Theta(\overline{J} \cdot V \sum_{n'=1}^{n} |\mathcal{M}_{n'}|)$

Debería quedar claro que cuanto más grande sea el tamaño de ventana, mayor número de rechazos de eventos n-grama habrá, por lo que el transductor resultante será más pequeño. Sin embargo, los resultados de traducción no se verán afectados ya que estas transiciones eliminadas son inalcanzables

por medio de dicho conjunto de test. Por tanto, cuanto mayor sea el tamaño de ventana, el transductor finalmente obtenido, $\mathcal{T}'$, estará más cerca de ser el subtransductor mínimo de $\mathcal{T}$ (considerando a $\mathcal{T}$ como el transductor que se obtiene sin podar) cuya relación asociada contiene aquellos pares de $R(\mathcal{T})$ cuya cadena de entrada pertenece al conjunto de muestras de test a traducir.

En la próxima sección, analizamos la construcción de símbolos extendidos a partir de un corpus bilingüe, clasificando los modelos derivados de GIATI según las restricciones sobre la cardinalidad de entrada en los símbolos de Γ.

3.3. Tipos de transductores GIATI

Tal y como ya se ha mencionado, la inferencia de transductores se realizará por medio de la transformación del corpus de entrenamiento bilingüe en un corpus de símbolos extendidos del que se aprenderá un modelo de lenguaje. Esta transformación se basará en la función de alineamiento definida para cada par de frases paralelas, por lo que, según el grado de alineamiento, los modelos estocásticos que finalmente se obtengan se podrán clasificar como transductores de estados finitos basados en palabras o basados en segmentos.

Por un lado, las funciones de alineamiento uno-a-uno y uno-a-muchos producirían modelos basados en palabras, mientras que, en cambio, las correspondencias muchos-a-muchos conducirían a modelos basados en segmentos.

3.3.1. Transductores basados en palabras de entrada

Si, por un lado, los modelos uno-a-uno no parecen una aproximación muy apropiada en traducción (debido a que, entre otras cosas, requieren que todos los pares de frases paralelas tengan exactamente el mismo número de palabras), los modelos de alineamiento uno-a-muchos, sin embargo, han sido un referente en traducción automática estadística hasta que las tendencias actuales basadas en segmentos ocuparon su lugar en la comunidad científica. Los modelos basados en palabras restringen la capacidad de alineamiento

entre un par de frases bilingües de manera que cada palabra de salida puede estar alineada como máximo con una de las palabras de entrada. En el otro sentido, y hablando de relaciones uno-a-muchos, el número de palabras de salida con las que las palabras de entrada pueden alinearse no está limitado.

Los modelos basados en palabras restringen las segmentaciones bilingües de modo que la longitud de todos los segmentos de entrada es de 1 palabra. Por lo tanto, los símbolos del alfabeto extendido Γ serán de la forma $(\mathbf{s}_j, \bar{t}_j)$, determinando de este modo el número de segmentos K de cada segmentación (o la longitud de $\overline{\gamma}$) a partir de la longitud de la frase de entrada $\mathbf{s} = \mathbf{s}_1 \dots \mathbf{s}_J$. Así, la cadena de símbolos extendidos $\overline{\gamma}$ se construye de la siguiente manera:

$$K = J$$

$$\mathbf{s} = \mathbf{s}_1 \dots \mathbf{s}_K$$
$$\mathbf{t} = \bar{t}_1 \dots \bar{t}_K$$

$$\overline{\gamma} = \gamma_1 \cdots \gamma_K$$

de modo que $\forall k = 1 \dots K \ : \ \gamma_k = (\mathbf{s}_k, \bar{t}_k) \in \Sigma \times \Delta^\star$.

De hecho, los transductores basados en palabras que se describen en esta sección se pueden considerar un caso particular de los basados en segmentos, condicionando la longitud de los segmentos de entrada a un único elemento. No obstante, el hecho de poder alinear una palabra de entrada dada con la cadena vacía ε es una de las diferencias entre esta aproximación y la descrita en la sección 3.3.2, en la que cada segmento de entrada debe alinearse forzosamente con un segmento de salida, impidiendo su alineamiento con ε.

La conversión de cada par de frases bilingües en una cadena de símbolos extendidos se realiza por medio de un algoritmo de etiquetado que, a cada palabra de entrada, le asigna una secuencia ordenada de palabras de salida, basada en su conjunto de alineamiento (y en el de las palabras anteriores), respetando el orden de aparición de éstas en la frase de salida. Esta condición puede provocar que a una palabra de entrada se le asigne la cadena vacía, teniendo que ver retrasada la aparición de las palabras con las que se alinea.

El método que presentamos en el algoritmo 3.2, que no es la única opción que conduce a los así llamados transductores GIATI basados en palabras, persigue que este retraso, en caso de que así suceda, sea mínimo, asignando las palabras de salida tan pronto como los alineamientos lo permitan.

Algoritmo 3.2 Generación de símbolos extendidos basados en palabras

Datos
  $\mathbf{s} = \mathbf{s}_1\,\mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$
  $\mathbf{t} = \mathbf{t}_1\,\mathbf{t}_2 \ldots \mathbf{t}_I \in \Delta^\star$
  $a = a_1 a_2 \ldots a_I$
Resultado $\overline{\gamma} = \gamma_1\gamma_2 \ldots \gamma_J \in \Gamma^\star$
Método
  $i = 1$
  $\forall j = 1 \ldots J$
    $\gamma_j = \mathbf{s}_j$
    mientras $((i \leq I) \wedge (a_i \leq j))$
      $\gamma_j = \gamma_j\,\mathbf{t}_i$
      i++
  mientras $(i \leq I)$
    $\gamma_J = \gamma_J\,\mathbf{t}_i$
    i++
Coste: $\Theta(J + I)$

Lo que significa que los símbolos extendidos se generan de izquierda a derecha, y en los que una palabra destino $\mathbf{t}_i$ se concatena con su correspondiente palabra fuente $\mathbf{s}_{a_i}$ si y sólo si su alineamiento $\mathbf{t}_i \rightarrow \mathbf{s}_{a_i}$ no se cruza espacialmente con ningún otro alineamiento que no se haya explorado aún. Si esta acción no es posible, entonces la aparición de $\mathbf{t}_i$ se retrasa hasta que la variable j haya alcanzado la última posición implicada en el grupo de alineamientos cruzados. Las palabras de aparición espontánea se sitúan en sus posiciones correspondientes, dado que es imperativo que todas las palabras respeten un riguroso orden monótono. Este procedimiento asegura que cada símbolo extendido está compuesto estrictamente de una palabra de entrada, opcionalmente seguida por un número arbitrario de palabras de salida. Como ejemplo, el alineamiento de la figura 3.5 exige que la aparición de $\mathbf{t}_2$ (y

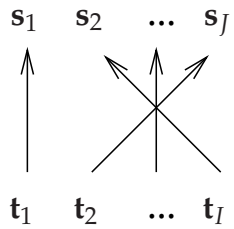


Figura 3.5: Un alineamiento basado en palabras

siguientes) tenga que demorarse hasta la generación de la palabra $\mathbf{s}_J$, lo que requiere dejar en blanco la asignación de las palabras anteriores $\mathbf{s}_2 \cdots \mathbf{s}_{J-1}$, produciendo la cadena extendida $\overline{\gamma} = (\mathbf{s}_1, \mathbf{t}_1)\ (\mathbf{s}_2, \varepsilon)\ (\mathbf{s}_3, \varepsilon) \ldots (\mathbf{s}_J, \mathbf{t}_2\, \mathbf{t}_3 \ldots \mathbf{t}_I)$. Si se desea una descripción más detallada sobre esta función, véase [CVP05].

Como cada unigrama, bigrama, etc., se representa mediante una transición etiquetada mediante su último símbolo, y dado que cada uno de estos símbolos extendidos se componen exactamente de una única palabra de entrada, la función de etiquetado inverso se puede aplicar de manera directa. De este modo, las etiquetas de transición se transforman de nuevo en pares de palabras de entrada y de salida para convertirse en las de un transductor.

3.3.2. Transductores basados en segmentos

Existen trabajos en los que también se emplea una metodología basada en segmentos en un marco de trabajo de estados finitos [CV04, CVP05, KDB06].

Por un lado, [CV04, CVP05] propone una versión de GIATI basada en segmentos, si bien en unos casos sólo es posible el uso de unigramas como modelo de lenguaje (ya que se proporciona el vocabulario de pares de segmentos bilingües, pero no sus posibles secuencias), en otros casos las segmentaciones derivadas se obtienen a través de otras funciones de etiquetado.

Por otro lado, en [KDB06], se aplica un proceso de traducción generativo compuesto de varios modelos de traducción. Cada distribución constituyente

del modelo, incluyendo algunos aspectos tan conocidos en traducción estadística como por ejemplo el reordenamiento de segmentos o la inserción de palabras espontáneas, se implementa por medio de un transductor de estados finitos. Sin embargo, la metodología GIATI trata de combinar todas estas operaciones sobre un único modelo de traducción. Otros autores [PTC07, PTC08] también han adaptado la extensión de palabras a segmentos con respecto al aprendizaje de transductores estocásticos de estados finitos mediante GIATI.

Los transductores GIATI basados en segmentos provienen del concepto de segmentación bilingüe monótona, de manera que las cadenas de símbolos extendidos se componen de las secuencias ordenadas correspondientes de pares de segmentos bilingües alineados recíproca y monótonamente entre sí. La restricción que se introduce en este caso hace referencia a la imposibilidad de que haya segmentos vacíos en algún lado de la segmentación bilingüe, esto es, que las funciones de segmentación μ y ρ son estrictamente crecientes:

$$\forall k = 1 \dots K : \mu_k > \mu_{k-1} \wedge \rho_k > \rho_{k-1}$$

Así, al impedir que la cadena vacía ε se pueda considerar un segmento dado, el número máximo de segmentos de una segmentación bilingüe se determina mediante la menor de las longitudes de las dos cadenas implicadas, es decir:

$$0 < K \leq \min(J, I)$$

$$\mathbf{s} = \overline{s}_1 \dots \overline{s}_K$$
$$\mathbf{t} = \overline{t}_1 \dots \overline{t}_K$$

$$\overline{\gamma} = \gamma_1 \dots \gamma_K$$

de manera que $\forall k = 1 \dots K \ : \ \gamma_k = (\overline{s}_k, \overline{t}_k) \in \Sigma^+ \times \Delta^+$.

A continuación, analizaremos dos métodos para la obtención de cadenas de símbolos extendidos basadas en segmentos a partir de un corpus paralelo. Por un lado, una de las técnicas explota el uso de alineamientos a nivel de palabra para extraer segmentaciones bilingües y monótonas de las frases del corpus de entrenamiento. Por otro lado, se puede aprender un modelo de

traducción basado en segmentos e integrarlo en un sistema de traducción automática que nos permita seleccionar secuencialmente aquellos pares de segmentos del modelo que mejor se aproximan a las frases de entrenamiento.

Segmentación bilingüe y monótona mediante aglutinamiento mínimo

Este método de segmentación se caracteriza por la aplicación de ciertos heurísticos sobre un conjunto de alineamientos a nivel de palabra para así obtener una segmentación bilingüe y monótona del corpus de entrenamiento. Los criterios a aplicar permitirían, por ejemplo, condicionar la longitud de los segmentos obtenidos, considerando entonces como algo razonable que éstos estuvieran compuestos del menor número de palabras posible, tanto en la entrada como en la salida. Por lo tanto, el vocabulario del modelo de lenguaje se reduciría, dando lugar a una mejor estimación de sus parámetros. De este modo, estaríamos llevando el concepto de segmentación hacia su nivel máximo de expresión, de acuerdo a los alineamientos proporcionados.

Por ejemplo, supongamos el alineamiento entre cadenas de la figura 3.6, representado a nivel de palabra como una correspondencia uno-a-muchos.

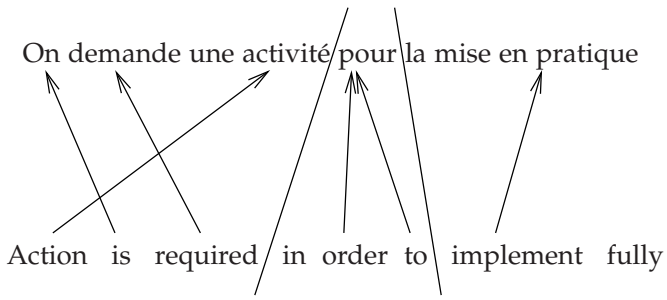


Figura 3.6: Segmentación monótona compatible con su alineamiento

Los alineamientos restringen el procedimiento de segmentación con el objetivo de obtener una secuencia monótona de segmentos bilingües de manera que dichos pares de segmentos sean los más pequeños posibles que mantengan internamente los alineamientos entre las palabras de entrada y de salida.

La figura indica asimismo los puntos de corte producto de la segmentación, cuya metodología, quizá sin optimizar, se puede observar en el algoritmo 3.3.

Algoritmo 3.3 Generación de símbolos extendidos basados en segmentos

Datos
$\mathbf{s} = \mathbf{s}_1\,\mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$
$\mathbf{t} = \mathbf{t}_1\,\mathbf{t}_2 \ldots \mathbf{t}_I \in \Delta^\star$
$a = a_1 a_2 \ldots a_I$
Resultado $\overline{\gamma} = \gamma_1\gamma_2 \ldots \gamma_K \in \Gamma^\star$
Variables $B_1^J, E_1^J : \{0, \ldots, I+1\}$
Método
$\forall j = 1 \ldots J : B_j = \min(I+1, \min_{a_i=j} i),\ E_j = \max(0, \max_{a_i=j} i)$
$\forall j = 1 \ldots J$
  $\forall j' = j+1 \ldots J$ si $(E_j \geq B_{j'})$
    $\forall j'' = j \ldots j' : B_{j''} = \min(B_j, B_{j'}),\ E_{j''} = \max(E_j, E_{j'})$
$\forall j = 1 \ldots J$ si $(E_j = 0)$
  $B_j = \max(1, \max(B_{j-1}, \min_{j'>j, B_{j'} \neq I+1} B_{j'}))$
  $E_j = \min(I, \max(E_{j-1}, \min_{j'>j, E_{j'} \neq 0} E_{j'}))$
$i = k = 1$
$\forall j = 1 \ldots J$
  $\gamma_k = \mathbf{s}_j$
  $j' = j$+1
  mientras $((j' \leq J) \wedge (B_{j'} = B_j) \wedge (E_{j'} = E_j))$
    $\gamma_k = \gamma_k\,\mathbf{s}_{j'}$
    j'++
  mientras $(i \leq E_j)$
    $\gamma_k = \gamma_k\,\mathbf{t}_i$
    i++
  $j = j' - 1,\ K = k,\ k$++
Coste: $\Theta(J^2)$

De este modo, se obtiene una segmentación bilingüe y monótona ($K = 3$) que ha conseguido identificar la siguiente correspondencia entre segmentos:

On demande une activité	*pour*	*la mise en pratique*
↕	↕	↕
Action is required	*in order to*	*implement fully*

Por lo tanto, la cadena de símbolos extendidos correspondiente estará compuesta de la secuencia monótona de los tres pares de segmentos bilingües, donde cada pareja de segmentos se debe considerar un símbolo individual en el contexto de la estimación de n-gramas como modelo de lenguaje:

S = { . . . , (On demande une activité, Action is required)
(pour, in order to) (la mise en pratique, implement fully), . . . }

Finalmente, una vez aprendido un autómata a partir del corpus S, éste se puede convertir en un transductor si consideramos cada símbolo extendido no como un elemento indivisible sino como el par de segmentos bilingües de entrada y salida que constituyen las etiquetas de transición en un transductor. En [CV04, GC07] se puede encontrar una descripción detallada acerca de la expansión de las transiciones basadas en segmentos a través de sus palabras de entrada constituyentes. La aplicación de la función de etiquetado inverso sobre una arista procedente del ejemplo propuesto se expone en la figura 3.7.

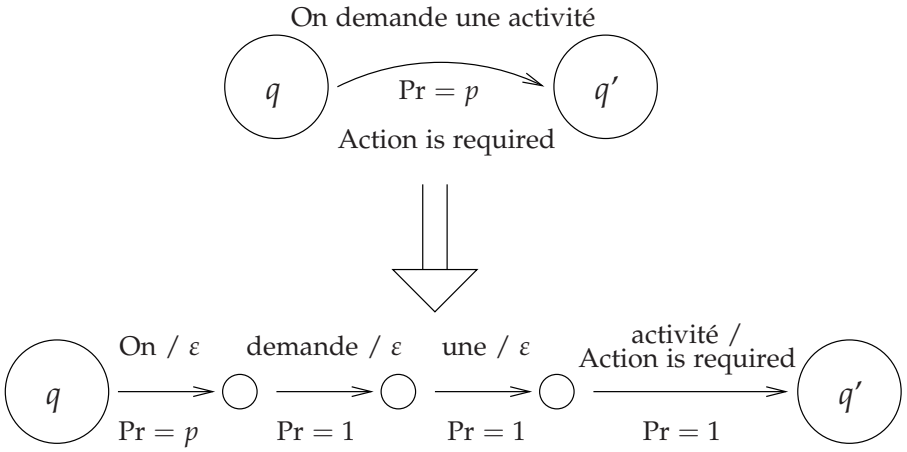


Figura 3.7: Una función de etiquetado inverso basada en segmentos

A continuación, presentamos un método de segmentación aproximado

por medio de los pares de segmentos bilingües de un modelo de traducción. A partir de ahí, el proceso de inferencia es el mismo que se acaba de describir.

Segmentación bilingüe y monótona usando otros modelos de traducción

A pesar de representar el estado del arte, el uso práctico de la mayoría de los sistemas de traducción basados en segmentos se encuentra muy limitado. La obtención de los mejores resultados de traducción se realiza a menudo mediante la integración de un gran número de modelos estadísticos distintos, lo que fácilmente puede llegar a ser una combinación de 8 modelos de media, que incluso en ocasiones puede extenderse con algún modelo más [KHB+07]. Resulta evidente que la gestión de tanta información durante la búsqueda tiene que llevar al sistema a un alto consumo de recursos computacionales, lo que repercutirá muy probablemente en el tiempo de respuesta del sistema.

Por otro lado, un marco de trabajo de estados finitos representa una alternativa interesante porque constituye un paradigma eficiente en el que factores de calidad y tiempo real se integran correctamente para construir dispositivos de traducción que puedan ser de ayuda a sus potenciales usuarios [CCM+94].

Debido al discreto éxito que los sistemas de traducción automática están obteniendo cuando se emprenden tareas de una cierta envergadura [KM06], los escenarios de trabajo interactivo se han vuelto muy populares [CCC+08]. La traducción asistida por computador (del inglés *Computer Assisted Translation*, CAT) se ha revelado en la última década como una potente interfaz entre usuarios y sistemas de traducción automática [BBC+09]. Ni qué decir tiene que el rendimiento a tiempo real es uno de los requisitos para una utilización productiva de cualquier marco de trabajo que se autodefina como interactivo.

GIATI, tal y como se ha definido en la sección 3.2, es un marco general para el diseño de algoritmos de inferencia de transductores. Describamos un algoritmo basado en modelos de traducción de segmentos bajo esta filosofía.

Tal y como se ha visto en la sección 2.2.2, un modelo de traducción basado en segmentos constituye un diccionario estocástico de segmentos bilingües.

De manera intuitiva, todos estos pares de segmentos se han obtenido a partir del cálculo de varias segmentaciones bilingües del corpus de entrenamiento. Sin embargo, para una aplicación estricta de la técnica GIATI, sólo se debería tener en cuenta una segmentación por cada par de frases, lo que implicaría tener que reducir este modelo mediante la selección de aquellos pares que componen esta única segmentación, descartando al resto de segmentos. Dicho requisito se aproxima eficientemente por medio de la traducción[1] de las frases de entrada del corpus de entrenamiento a través de un sistema de traducción automática basado en segmentos, ya que la búsqueda en modelos de traducción de segmentos implica la obtención de la mejor segmentación.

El empleo de esta técnica se puede formalizar por medio de un sistema monótono de traducción automática que usa un diccionario de segmentos $\mathcal{M}$ como modelo de traducción: para cada frase de entrada $\mathbf{s}$ perteneciente a la muestra, la cadena de símbolos extendidos se construye a partir de una secuencia de $K_{\mathbf{s}}$ pares de segmentos, $(\overline{s}_k, \overline{t}_k) \in \mathcal{M}$, en la que $\overline{s}_1 \overline{s}_2 \ldots \overline{s}_{K_{\mathbf{s}}} = \mathbf{s}$.

La segmentación como tal sólo se produce sobre la cadena de entrada, por lo que el número máximo de segmentos se acota a través de la longitud de $\mathbf{s}$:

$$0 < K \leq J$$

$$\begin{array}{c} \mathbf{s} = \overline{s}_1 \ldots \overline{s}_K \\ \quad\; \overline{t}_1 \ldots \overline{t}_K \end{array}$$

$$\overline{\gamma} = \gamma_1 \ldots \gamma_K$$

de manera que $\forall k = 1 \ldots K \; : \; \gamma_k = (\overline{s}_k, \overline{t}_k) \in \mathcal{M} \subset \Sigma^+ \times \Delta^+$.

Por lo tanto, el corpus de cadenas que se va a modelar mediante un autómata de estados finitos estará compuesto de las secuencias de pares de segmentos que mejor se ajustan a las frases de entrada del corpus de entrenamiento, según el sistema de traducción automática. Esto se puede ver como una reducción efectiva de la tabla de traducción de segmentos por medio de la selección de aquellos pares de segmentos que se corresponden con

[1] La búsqueda se debe restringir para encontrar una solución monótona

la segmentación más probable (junto con su traducción) de cada una de las frases de entrada del corpus de entrenamiento. Un esquema de este método de exportación de modelos de segmentos queda reflejado en la figura 3.8.

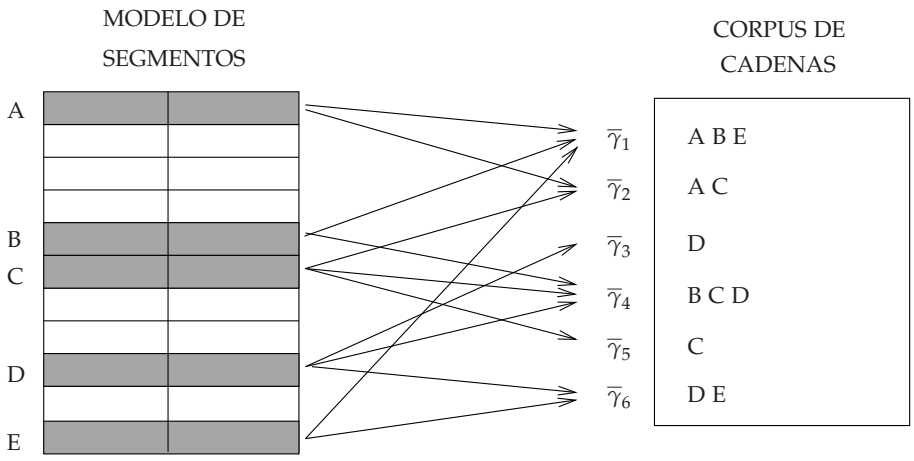


Figura 3.8: Usando modelos de traducción basados en segmentos bajo GIATI. Ejemplo aplicado sobre un corpus hipotético compuesto de 6 pares de frases. La función de etiquetado selecciona los segmentos del modelo de traducción, aproximando de forma subóptima la segmentación monótona más probable.

Basicamente, el algoritmo produce un transductor que incluye una cantidad más pequeña de pares bilingües de traducción, es decir, no todos los segmentos de $\mathcal{M}$, pero con un modelo probabilístico sobre el orden de éstos.

3.4. Descodificación utilizando modelos GIATI

La ecuación $\hat{\mathbf{t}} = \underset{\mathbf{t}\in\Delta^\star}{\operatorname{argmax}} F_{\mathcal{T}}(\mathbf{s},\mathbf{t})$ refleja el problema de la traducción automática en los términos de un transductor estocástico de estados finitos $\mathcal{T}$ cuya función $F_{\mathcal{T}}(\mathbf{s},\mathbf{t})$ es un modelo de $\Pr(\mathbf{s},\mathbf{t})$ o de $\Pr(\mathbf{t}\,|\,\mathbf{s})$. Dado que sólo se conoce la frase de entrada, el transductor se debe explorar en busca de todas aquellas frases de salida con las que se relaciona según el modelo.

Dada una frase de entrada, y según la aproximación polinómica a la búsqueda descrita en la sección 2.1.5, la mejor hipótesis de salida es aquella que se corresponde con un camino a través del modelo de transducción que, con mayor probabilidad, acepta la secuencia de entrada como parte del modelo de lenguaje del transductor. Aunque la exploración del modelo se restringe por medio de la frase de entrada, el espacio de búsqueda puede ser extremadamente grande. Como consecuencia, sólo las hipótesis de mayor puntuación parcial se consideran candidatas a convertirse en la solución del problema. Este proceso de búsqueda se lleva a cabo eficientemente por medio de una versión adaptada del conocido algoritmo de Viterbi con búsqueda en haz, cuyo coste asintótico temporal depende de la longitud de la frase a analizar y del tamaño del modelo, expresado mediante su número de estados y aristas.

3.4.1. Estrategias de búsqueda: de palabras a segmentos

La estructura en forma de retículo que se emplea habitualmente para analizar una frase de entrada a través de un modelo de estados finitos se puede limitar por medio de lo que se conoce como búsqueda en haz. Así, en cada etapa del análisis, sólo se tienen en cuenta los caminos de mayor puntuación.

Una estrategia de análisis basada en palabras partiría del estado inicial <s>, buscando las mejores transiciones compatibles con la primera palabra de entrada. Los correspondientes estados destino se introducirían en la estructura de salida, la cual se utilizaría para el análisis de la segunda palabra. Iterativamente, cada estado del *trellis* se examina para encontrar las transiciones cuyo símbolo de entrada se corresponde con la palabra actual de análisis, generando la siguiente etapa a partir de los estados destino mejor puntuados. Finalmente, los estados alcanzados en la última etapa de análisis se repuntúan de acuerdo a las probabilidades correspondientes de ser estados finales. Un esquema de esta estrategia de búsqueda se encuentra en el algoritmo 3.4.

Este es el algoritmo estándar para analizar una frase de entrada a través de un modelo estocástico de estados finitos. Sin embargo, puede no ser el más apropiado cuando el modelo es un transductor de los basados en segmentos.

Algoritmo 3.4 Estrategia de búsqueda basada en palabras

Datos
$\mathbf{s} = \mathbf{s}_1\, \mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$
$\mathcal{T} = (\Sigma, \Delta, Q, i, f, P)$ con un Σ-indeterminismo medio de N
Resultado $\hat{\mathbf{t}} \in \Delta^\star$
Variables $C_1^{|Q|}, C_1'^{|Q|} : \mathbb{R}$, $\mathbf{t}_1^{|Q|}, \mathbf{t}_1'^{|Q|} \in \Delta^\star$
Método
$C'_{<s>} = 1,\ \mathbf{t}'_{<s>} = \varepsilon$
$\forall j = 1 \ldots J$
$\forall w \in C'$
$\forall (w, \mathbf{s}_j, \bar{t}, q) \in P$
si $(C'_w \cdot P(w, \mathbf{s}_j, \bar{t}, q) > C_q)$
$C_q = C'_w \cdot P(w, \mathbf{s}_j, \bar{t}, q)$
$\mathbf{t}_q = \mathbf{t}'_w\, \bar{t}$
$C' = C$
$\mathbf{t}' = \mathbf{t}$
$\hat{q} = \underset{q \in F}{\text{argmax}} \, [\, C_q \cdot f(q)\,]$
$\hat{\mathbf{t}} = \mathbf{t}_{\hat{q}}$
Coste: $\Theta(J \cdot \overline{|C|} \cdot N)$

Como se puede apreciar en la figura 3.7, una serie de transiciones consecutivas representan una única probabilidad de traducción entre segmentos, a partir de una cierta historia. De hecho, el camino entre los estados q y q' sólo se debería considerar en caso de que la secuencia de entrada pendiente de análisis comenzara exactamente con el segmento *On demande une activité*. Si no es así, dicha ruta no debería tenerse en cuenta.

Sin embargo, mientras las palabras de entrada vayan siendo compatibles con las correspondientes aristas, y en función de su probabilidad de transición, este algoritmo de análisis sincronizado a nivel de palabra puede considerar estos estados intermedios durante el proceso de búsqueda, incluso aunque finalmente la ruta del segmento no se llegue a completar. Como consecuencia, estos estados estarán ocupando una posición valiosa y a la vez inútil dentro de la estructura del *trellis*.

Especialmente importante para determinar el alcance de una búsqueda

restringida mediante haz dinámico es el vértice de máxima probabilidad acumulada dentro de cada etapa. Si dicho vértice representa el análisis parcial de un segmento incompleto, entonces el alcance efectivo del haz puede verse reducido significativamente, sobre todo si éste es relativamente pequeño, provocando el rechazo de algunas otras opciones reales de análisis debido, a priori, a su menor puntuación.

Por tanto, esta estrategia de búsqueda puede llevar al sistema a disminuir la calidad de sus traducciones. Alternativamente, la solución pasa por incrementar el tamaño del haz para poder tener en cuenta los caminos de éxito en el proceso de búsqueda, por lo que el tiempo de descodificación aumentaría.

En el caso de una búsqueda restringida mediante haz estático, la situación es todavía más grave debido a que el número de posiciones por etapa se fija previamente, de modo que cada una representa una posibilidad de análisis. Cualquiera de ellas que esté albergando a un estado que sea, o que conduzca a un sumidero, reduce de manera efectiva el alcance del haz de exploración.

Por otro lado, una estrategia de búsqueda basada en segmentos nunca incluiría estados intermedios en el *trellis*. En su lugar, estos caminos se intentan recorrer hasta que se alcanza un estado de la topología original del autómata, como q' en la figura 3.7, lo que da lugar a la aparición de arcos en el grafo multietapa asociado cuyos vértices, situados en etapas no consecutivas, distan entre sí tantas etapas como número de palabras de entrada tengan los segmentos correspondientes a las secuencias de transiciones del transductor.

En la figura 3.9, se puede ver un grafo multietapa asociado a una estrategia de búsqueda basada en segmentos. Sin embargo, en la práctica, el *trellis* se construye de izquierda a derecha utilizando un par de vectores columna que se desplazan sobre todas las etapas consecutivas de la matriz dos a dos. La generación de estados en cada etapa se realiza en función de su alcanzabilidad a partir de los estados ya existentes en dicha etapa o de los de la etapa anterior, lo que, para las transiciones basadas en segmentos, se traduce en un adelantamiento de los nodos destino desde su propia etapa hasta la presente.

Nótese que, aunque parezca un retorno a la estrategia de búsqueda basa-

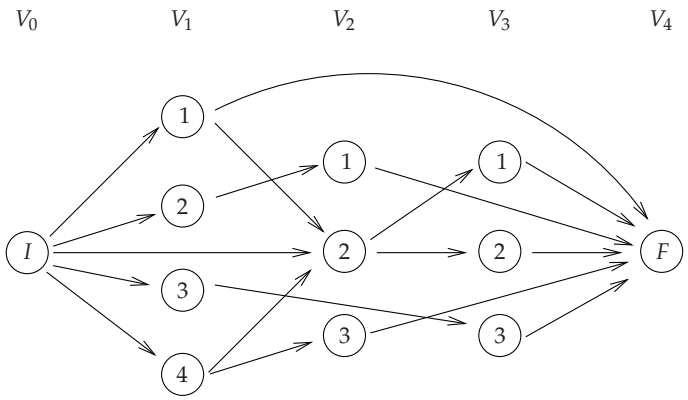


Figura 3.9: Grafo multietapa con arcos o transiciones basadas en segmentos. La arista entre el estado $1 \in V_1$ de la etapa 1 y el estado $F \in V_4$ de la etapa 4 representa una transición mediante un segmento de tres palabras de entrada.

da en palabras, la diferencia estriba en que, mientras en aquélla los estados que se introducían en el *trellis* son los llamados estados intermedios, que no pertenecen a la topología del autómata, en ésta los estados del *trellis* son los estados alcanzables tras haber analizado una secuencia completa de palabras.

Por lo tanto, los estados del *trellis* se deben almacenar junto con un indicador que refleje la última posición analizada dentro de la frase de entrada. Como ejemplo, la secuencia $I - 1 - F$ de la figura 3.9 se implementa en la práctica como $I_0 - 1_1 - F_4 - F_4 - F_4$, tal como la figura 3.10 pretende mostrar.

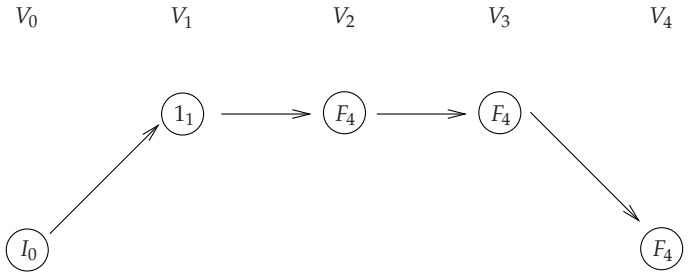


Figura 3.10: Desarrollo de las transiciones basadas en segmentos en el *trellis*. Cada nodo indica la etapa a partir de la que se tiene que continuar el análisis.

Así, en cada iteración, los estados cuyo indicador de posición sea inferior al de la etapa actual de análisis se examinan en busca de aristas compatibles. En caso contrario, dichos estados ya contemplan la palabra correspondiente por lo que se transfieren directamente a la siguiente etapa sin cambio alguno. El código de esta otra estrategia de búsqueda se describe en el algoritmo 3.5.

Algoritmo 3.5 Estrategia de búsqueda basada en segmentos

Datos
  $\mathbf{s} = \mathbf{s}_1\,\mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$
  $\mathcal{T} = (\Sigma, \Delta, Q_\mathcal{A} \cup Q_\mathcal{T}, i, f, P)$ con un Σ-indeterminismo medio de N
Resultado $\hat{\mathbf{t}} \in \Delta^\star$
Variables $C_1^{|Q|}, C_1'^{|Q|} : \mathbb{R}$, $\mathbf{t}_1^{|Q|}, \mathbf{t}_1'^{|Q|} \in \Delta^\star$, $L_1^{|Q|} : \{1, \ldots, J\}$
Método
  $C'_{<s>} = 1,\ \mathbf{t}'_{<s>} = \varepsilon,\ L_{<s>} = 0$
  $\forall j = 1 \ldots J$
    $\forall w \in C'$
      si $(L_w \geq j)$
        $C_w = C'_w$
        $\mathbf{t}_w = \mathbf{t}'_w$
      si no $\forall (w, \mathbf{s}_j, \bar{t}, q) \in P$
        $tmp = C'_w \cdot P(w, \mathbf{s}_j, \bar{t}, q)$
        $j' = j$+1
        mientras $((q \notin Q_\mathcal{A}) \wedge ((q, \mathbf{s}_{j'}, \bar{t}, q') \in P))$
          $q = q'$
          j'++
        si $((q \in Q_\mathcal{A}) \wedge (tmp > C_q))$
          $C_q = tmp$
          $\mathbf{t}_q = \mathbf{t}'_w\,\bar{t}$
          $L_q = j'$-1
    $C' = C$
    $\mathbf{t}' = \mathbf{t}$
  $\hat{q} = \underset{q \in F}{\text{argmax}}\ [\ C_q \cdot f(q)\ ]$
  $\hat{\mathbf{t}} = \mathbf{t}_{\hat{q}}$
Coste: $\Theta(J \cdot \overline{|C|} \cdot N)$

El algoritmo se mantiene síncrono a nivel de palabra, pero, en esta ocasión,

lo que se garantiza es que los vértices de la j-ésima etapa han analizado al menos hasta la j-ésima palabra, pero quizá pueden haber llegado más lejos. También se garantiza que todos los estados de las sucesivas etapas del *trellis*, como pertenecen a la topología original del autómata, conducen a un estado final del modelo, debido a que, o bien son estados finales por sí mismos, o bien existe una secuencia de transiciones *backoff* que llevan a un estado final.

3.4.2. Suavizado mediante *backoff*

En la sección 3.1.1, se introdujo la necesidad de disponer de un mecanismo de suavizado en modelado del lenguaje, y particularmente en los n-gramas, ya que el lenguaje natural presenta un alto grado de dispersión en sus datos, lo que ocasiona que los modelos sean incapaces de capturar toda la casuística existente a través de una simple muestra, por muy voluminosa que ésta sea. Los modelos entrenados contemplan esta posibilidad por medio de una serie de transiciones de *backoff* que conducen hacia estados de capas inferiores junto con la reserva de cierta probabilidad para los elementos desconocidos.

Para analizar con eficacia las frases de entrada a través de los modelos GIATI se han explorado dos criterios distintos para el comportamiento del suavizado de los modelos por medio de las aristas de *backoff*. En primer lugar, se ha implementado la extrapolación del algoritmo de *backoff* desde el dominio de los autómatas al de los transductores a partir de su propia definición en los modelos de n-gramas en la que una arista de *backoff* se considera una transición de fallo. Es decir, a partir de un cierto estado q, sólo si no ha habido ninguna transición de éxito compatible con la entrada, entonces se habilitaría su transición de *backoff* con destino al nivel inferior. Lamentablemente, aunque el algoritmo de suavizado se mantenga, al haber cambiado el dominio de los lenguajes por el de las relaciones, la exploración estructural del modelo no es completamente equivalente, ignorando algunas relaciones del transductor cuya revelación implica el acceso a capas inferiores.

Sin embargo, es interesante destacar que nuestro modelo de traducción es en realidad un modelo de lenguaje cuyos símbolos individuales están forma-

dos por pares de secuencias bilingües, mientras que las frases de entrada contra las que usamos nuestro modelo no contienen símbolos de tal naturaleza. En su lugar, dichas frases (pertenecientes a $\Sigma^\star$) se pueden analizar como si se expandieran en todos aquellos pares de etiquetas bilingües del alfabeto Γ con los que son compatibles a un nivel de lenguaje de entrada.

Por consiguiente, la segunda interpretación para las aristas de *backoff* analiza la entrada como si ésta estuviese compuesta de símbolos bilingües, aplicando de este modo el criterio de suavizado anterior de manera individual sobre cada símbolo extendido del alfabeto Γ con el que sea compatible. Es decir, dada una transición de éxito etiquetada mediante cierto símbolo bilingüe, no hay ninguna necesidad de tomar la correspondiente transición de *backoff* para encontrar otros caminos etiquetados con el mismo símbolo, pero sí para tratar de encontrar otras transiciones cuya etiqueta sea cualquier otro símbolo del vocabulario extendido, también compatible con la entrada. Este comportamiento se puede implementar de forma eficiente considerando a los arcos de *backoff* como transiciones ε/ε manteniendo una lista dinámica de estados prohibidos cada vez que se desciende por una arista de *backoff*. Un esbozo sobre la gestión interna de activación y desactivación de estados, integrada en el propio proceso de exploración, se muestra en el algoritmo 3.6.

El algoritmo de búsqueda trata de traducir varias palabras consecutivas de entrada a través de una secuencia de transiciones que se corresponden con un símbolo extendido del modelo. A continuación, el proceso se repite iterativamente tras descender a través de la arista de *backoff* correspondiente con el objetivo de cubrir el resto de símbolos compatibles, no sólo los que existen tras una cierta historia, sino también tras cualquiera de sus sufijos. Con ayuda de la desactivación de estados, se puede cumplir esta restricción: cualquier camino entre dos estados q y q' debe trazarse a través del mínimo número de aristas de *backoff*; cualquier otro camino $q \rightarrow q'$ o $q \rightarrow q''$, donde q'' es el destino de una secuencia de transiciones de *backoff* desde q', se debe ignorar. La figura 3.11 muestra un ejemplo de análisis sobre un transductor estocástico basado en un modelo de bigramas suavizado por medio de *backoff*.

Algoritmo 3.6 Método de suavizado condicional basado en transiciones ε/ε

Datos
$\mathbf{s} = \mathbf{s}_1\,\mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$
$\mathcal{T} = (\Sigma(\Gamma), \Delta(\Gamma), Q, i, f, P)$ con una Σ-ambigüedad media en Γ de N
Resultado $\hat{\mathbf{t}} \in \Delta^\star$
Variables $C_1^{|Q|}, C_1'^{|Q|} : \mathbb{R}, \ldots$
Método
[...]
$\forall j = 1 \ldots J$
  $\forall w \in C'$
    [...]
    $\forall w \rightarrow q \in$ {transiciones}
      si q está activo
        [...]
        desactivar q
    si w no es ε
      si w no está en el nivel más alto
        $\forall q \in$ {estados inactivos}
          activar q
          desactivar $q' : (q, \varepsilon, \varepsilon, q') \in P$
      $w = q : (w, \varepsilon, \varepsilon, q) \in P$
    si no
      [...]
      $\forall q \in$ {estados inactivos} : activar q
[...]
Coste: $\Theta(J \cdot \overline{|C|} \cdot N)$

Esta segunda interpretación para las aristas de *backoff* de un transductor aplica un suavizado equivalente al del modelo de n-gramas en el que se basa, a diferencia de la primera, cuya implementación es una aproximación a éste.

3.5. Combinación log-lineal de transductores

El rendimiento de los sistemas de traducción automática se ha visto incrementado significativamente cuando se utilizan modelos adicionales en un en-

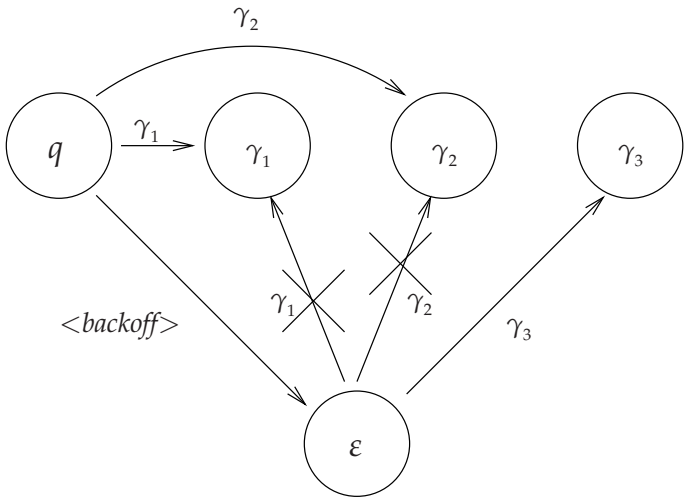


Figura 3.11: Aristas compatibles para un modelo suavizado de bigramas. Dado un estado alcanzable q, asumamos que los símbolos γ_1, γ_2, y γ_3, son todos compatibles con la secuencia de entrada todavía pendiente de análisis. Sin embargo, el bigrama $q\gamma_3$ no sucede a lo largo del corpus de entrenamiento, por lo tanto no hay una transición directa entre los estados q y γ_3. La arista de *backoff* habilita el acceso al estado γ_3 ya que el bigrama $q\gamma_3$ se convierte en el evento unigrama γ_3 que sí está contemplado en el modelo. Las transiciones unigrama a los estados γ_1 y γ_2 se deben ignorar ya que sus correspondientes eventos bigrama fueron encontrados en niveles superiores.

torno de modelado *log-lineal* [ON03]. Dichos modelos suelen ser típicamente: modelos de traducción basados en segmentos, modelos de lenguaje, modelos de longitud de las frases, diccionarios estocásticos, etc. Esta aproximación representa en la actualidad el estado del arte en traducción estadística [Lop08].

El objetivo de esta sección es el de introducir el marco apropiado para una aproximación log-lineal en el campo de los modelos de estados finitos en traducción automática estadística. Otros autores también han abordado aproximaciones log-lineal bajo formalismos de estados finitos [Eis02, DSE08].

En nuestro caso, este tipo de modelado se plantea por medio de la combinación e integración de diversos transductores estocásticos de estados finitos, permitiendo una representación sencilla y eficiente del estado del arte actual.

A pesar de que los modelos de estados finitos constituyen una opción interesante en traducción automática estadística, demostrada en diferentes ámbitos de traducción [CNO+04, GSC08], el empleo de modelos adicionales dentro de un marco de trabajo log-lineal no parece una tarea fácil a priori. Sin embargo, la utilización de los transductores basados en segmentos proporciona el entorno adecuado para facilitar en gran medida esta posibilidad.

Dichos transductores permiten una cómoda combinación de modelos estadísticos locales mediante la ampliación del número de funciones de probabilidad en las transiciones de los propios transductores. Cada modelo, pues, es capaz de otorgar una probabilidad distinta a cada par de segmentos bilingües según su propia naturaleza. La combinación de modelos estocásticos viene respaldada por un marco de trabajo log-lineal en el que, mediante un sencillo algoritmo de descenso por gradiente cuya función objetivo esté basada en medidas de evaluación automática, podemos encontrar la selección óptima de pesos que proporcionan un mejor rendimiento global del sistema. La integración de todos los modelos mediante una simple combinación lineal permite regresar al formalismo inicial donde las aristas tienen un único peso, que en este caso, no se puede garantizar que siga siendo una probabilidad, derivando en un transductor *ponderado* de estados finitos, que no estocástico, pero que modela la combinación log-lineal de diversos modelos que sí lo son. No obstante, los algoritmos de búsqueda no varían, pudiendo usar incluso los desarrollos previos sobre suavizado y sobre análisis basado en segmentos.

3.5.1. Modelos locales basados en segmentos

La denominación de modelo *local* basado en segmentos hace referencia a que la probabilidad correspondiente se modela como una operación matemática que comprende otras distribuciones de probabilidad de ámbito menor, en este caso asociadas sobre cierto conjunto de pares de segmentos bilingües.

Por ejemplo, los modelos de traducción que modelan $\Pr(\mathbf{s} \mid \mathbf{t})$ a un nivel de frase completa se expresan comúnmente como una serie de distribuciones de probabilidad de traducción entre pares de segmentos $\Pr(\mathbf{s}_j \dots \mathbf{s}_{j'} \mid \mathbf{t}_i \dots \mathbf{t}_{i'})$. De esta manera, como vimos en la sección 2.2.2, $\Pr(\mathbf{s} \mid \mathbf{t})$ se modela mediante el sumatorio de la probabilidad asociada a cada posible segmentación, cuyo cómputo individual se obtiene a partir del productorio de las probabilidades $\Pr(\mathbf{s}_j^{j'} \mid \mathbf{t}_i^{i'})$ de los pares de segmentos que intervienen en dicha segmentación.

Las probabilidades de traducción entre pares de segmentos se pueden expresar adecuadamente como transiciones cíclicas en un transductor estocástico que carece de sintáxis o estructura, con tantos ciclos como pares de segmentos bilingües haya en el modelo. La figura 3.12 muestra la representación por medio de transductores de modelos de traducción basados en segmentos.

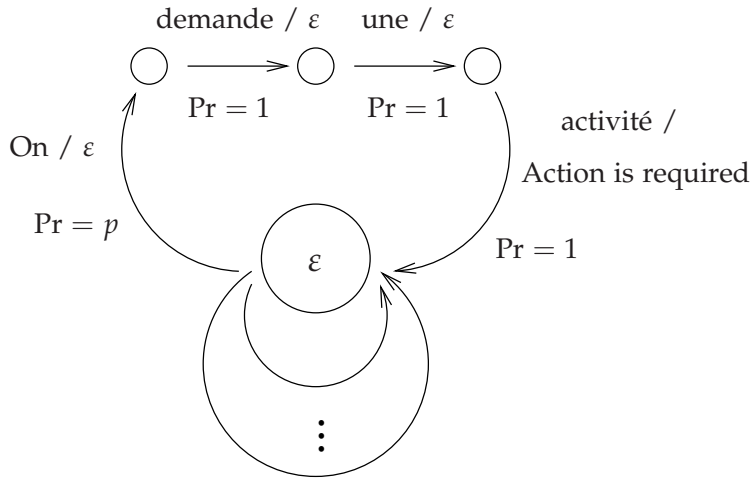


Figura 3.12: Un transductor basado en segmentos como modelo de traducción

La restricción de monotonicidad sobre los modelos de traducción basados en segmentos, planteada en la sección 2.2.2, permite su implantación en un marco de trabajo de estados finitos ya que dichos modelos son monótonos. Así, una búsqueda de Viterbi a través de un transductor de estas características modela $\Pr(\mathbf{s} \mid \mathbf{t})$ a partir de la segmentación monótona más probable.

Dicho así, se establecen de forma intuitiva las semejanzas entre los modelos de traducción monótonos, cuya expresión en términos de modelado del lenguaje se podría asimilar a los modelos de unigramas (que no tienen en cuenta la historia previa), y los transductores GIATI basados en segmentos, que modelan la historia reciente mediante modelos de orden superior, estimando así la probabilidad de cualquier segmentación bilingüe y monótona.

Para una combinación óptima de modelos locales y transductores GIATI, todos ellos deben compartir el vocabulario de pares de segmentos bilingües, de modo que, para todo $(\mathbf{s}_j^{j'}, \mathbf{t}_i^{i'})$, cada modelo tenga su probabilidad asociada.

En la práctica, cualquier modelo monótono cuya evaluación sobre $\mathbf{s}$ y $\mathbf{t}$ pueda descomponerse por medio de su segmentación bilingüe más probable es susceptible de ser utilizado en combinación local con transductores GIATI. Los modelos de traducción inversos, es decir, aquellos que modelan $\Pr(\mathbf{s} \mid \mathbf{t})$, son firmes candidatos a este tipo de combinación, como ya adelantábamos. Por supuesto, los modelos de traducción directos, donde se modela $\Pr(\mathbf{t} \mid \mathbf{s})$, también son válidos para ser integrados bajo esta aproximación log-lineal. Adicionalmente, los modelos de longitud de las frases es otra posibilidad.

3.5.2. Búsqueda log-lineal

Tal y como se ha descrito en la sección 2.1.5, el problema de la búsqueda a través de modelos de estados finitos se resuelve eficientemente por medio del algoritmo de descodificación de Viterbi. Se necesita entonces una estructura en forma de *trellis* para cada uno de los M modelos basados en segmentos, que permita analizar la entrada $\mathbf{s}$ y recopilar sus hipótesis correspondientes.

Sin embargo, la topología de los modelos a combinar es muy diferente. Mientras que, por un lado, la topología de los modelos GIATI presenta una cierta estructura, la de cualquier modelo local es relativamente muy sencilla, dado que, bajo una estrategia de análisis basada en segmentos (sección 3.4.1), en el contexto de los modelos de lenguaje, los autómatas tienen 1 único estado (el estado ε de la figura 3.12) y disponen del mismo conjunto de transiciones.

De este modo, y dado que todos los modelos comparten el vocabulario de pares de segmentos bilingües, el *trellis* que se construye para resolver el problema de búsqueda planteado en la ecuación (2.10) es algorítmicamente equivalente al que se hubiere construido únicamente para el modelo GIATI, salvo que, en esta ocasión, el coste asociado a cada arista del grafo multietapa no sólo depende de la probabilidad derivada de la transición correspondiente del transductor GIATI, sino también del peso λ de dicho modelo, y de la contribución asignada por cada modelo adicional sobre el mismo par de segmentos implicado en dicha transición, corregida mediante su propio peso.

Esta descomposición incremental del cálculo de la ecuación (2.10) se puede realizar debido a que el valor de la contribución asociada a cada modelo se aproxima por la del camino más probable entre todas sus derivaciones $d(\mathbf{s}, \mathbf{t})$, supuestamente analizado como una segmentación bilingüe en K segmentos, igual que sucede en el análisis mediante un único modelo de estados finitos:

$$\mathbf{s} = \bar{s}_1 \ldots \bar{s}_K \qquad\qquad \mathbf{t} = \bar{t}_1 \ldots \bar{t}_K$$

$$\forall m \in \{1, \ldots, M\} : h_m(\mathbf{s}, \mathbf{t}) \approx \sum_{k=1}^{K} h_m(\bar{s}_k, \bar{t}_k)$$

donde $h_m(\bar{s}_k, \bar{t}_k)$ es la aplicación del modelo m sobre dicho par de segmentos.

De este modo, la ecuación (2.10) se puede expresar finalmente como sigue:

$$\begin{aligned} \hat{\mathbf{t}} &= \operatorname*{argmax}_{\mathbf{t}} \sum_{m=1}^{M} \lambda_m \cdot h_m(\mathbf{s}, \mathbf{t}) \\ &= \operatorname*{argmax}_{\mathbf{t}} \sum_{m=1}^{M} \lambda_m \cdot \sum_{k=1}^{K} h_m(\bar{s}_k, \bar{t}_k) \\ &= \operatorname*{argmax}_{\mathbf{t}} \sum_{k=1}^{K} \left[\sum_{m=1}^{M} \lambda_m \cdot h_m(\bar{s}_k, \bar{t}_k) \right] \end{aligned} \tag{3.2}$$

lo que permite una construcción incremental de la expresión a maximizar por medio de la contribución de cada modelo (junto con su peso asociado) sobre el par de segmentos que supone la extensión de una hipótesis parcial. Por lo tanto, la contribución global de todos los modelos en cada arista del

trellis será de una puntuación parcial de $\sum_{m=1}^{M} \lambda_m \cdot h_m(\bar{s}_k, \bar{t}_k)$ para extender una hipótesis parcial a través de una transición basada en los segmentos $\bar{s}_k$ y $\bar{t}_k$. Esto se puede ver como una operación de recalificación o *rescoring* en la que las probabilidades del modelo GIATI se modifican adecuadamente debido a la influencia del resto de modelos, según la distribución de pesos λ existente.

Sin embargo, en lugar de representar cada modelo mediante un transductor independiente, en la práctica resulta mucho más cómodo utilizar una única estructura de modelado. Para ello, los modelos locales se incrustan dentro del transductor GIATI a través de la extensión de su función de transición P.

Un transductor extendido se puede definir entonces para que pueda tener en cuenta diversos valores de función para cada arista definida en el modelo:

$$P_1^M : Q \times \Sigma^\star \times \Delta^\star \times Q \rightarrow \mathbb{R}$$

donde, $\forall m \in \{1, \ldots, M\}$, P_m denotaría el m-ésimo valor de función de P. La figura 3.13 representa la estructura de un transductor extendido, con $M = 3$.

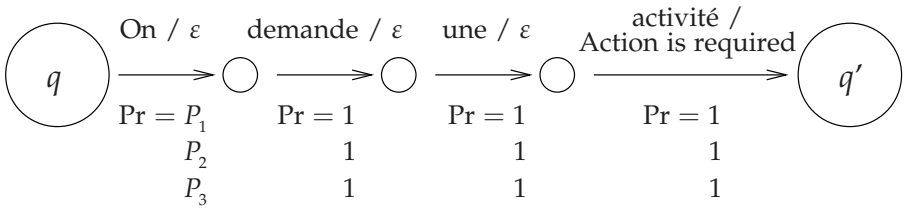


Figura 3.13: Modelos locales incrustados dentro de un transductor GIATI

Obviamente, esta integración de modelos causa un incremento sustancial de los requerimientos de memoria ya que las contribuciones de los modelos locales se repiten a lo largo de diversas transiciones del transductor GIATI. Sin embargo, esta representación nos permitirá mantener el algoritmo de búsqueda prácticamente inalterado, lo que compensa sobradamente el sacrificio. Como muestra, el procedimiento de búsqueda se describe en el algoritmo 3.7.

Algoritmo 3.7 Método de búsqueda log-lineal sobre transductores extendidos

Datos

$\mathbf{s} = \mathbf{s}_1\, \mathbf{s}_2 \ldots \mathbf{s}_J \in \Sigma^\star$

$\mathcal{T} = (\Sigma, \Delta, Q, i, f, P_1^M)$

$\lambda_1^M : \mathbb{R}$ es la distribución de pesos de la combinación log-lineal

Resultado $\hat{\mathbf{t}} \in \Delta^\star$

Variables $C_1^{|Q|}, C_1'^{|Q|} : \mathbb{R}$, $\mathbf{t}_1^{|Q|}, \mathbf{t}_1'^{|Q|} \in \Delta^\star$, ...

Método

```
[...]
```

$\forall j = 1 \ldots J$

  $\forall w \in C'$

```
    [...]
```

    $\forall (w, \mathbf{s}_j, \bar{t}, q) \in P$

```
      [...]
```

      $C_q = C'_w + \sum_{m=1}^{M} \lambda_m \cdot \log P_m(w, \mathbf{s}_j, \bar{t}, q)$

```
      [...]
    [...]
```

  $C' = C$

  $\mathbf{t}' = \mathbf{t}$

$\hat{q} = \underset{q \in F}{\operatorname{argmax}} \, [\, C_q + \lambda_1 \cdot \log f(q) \,]$

$\hat{\mathbf{t}} = \mathbf{t}_{\hat{q}}$

Coste: $\Theta(J \cdot \overline{|C|} \cdot N \cdot M)$ siendo N la exploración media por estado y etapa

Una vez el conjunto de parámetros $\hat{\lambda}_m$ se ha establecido a través de un conjunto de validación y de un algoritmo de optimización multidimensional, como por ejemplo el método *Downhill Simplex* [NM65], se puede obtener de nuevo un transductor al uso, equivalente al transductor log-lineal extendido, donde solamente exista un valor de función por cada transición del modelo. Esta operación se lleva a cabo mediante el cálculo de la combinación lineal entre los diversos valores de cada arco, ponderados con sus pesos óptimos. Por último, las transiciones cuyo peso se haya visto reducido a 0, por medio de la asignación nula por parte de alguno de los modelos, se pueden eliminar del transductor ya que representan aristas inexistentes en la práctica, redu-

ciendo de este modo el tamaño de los modelos, y por tanto también, la complejidad de los algoritmos que los manejan. La figura 3.14 ilustra este proceso.

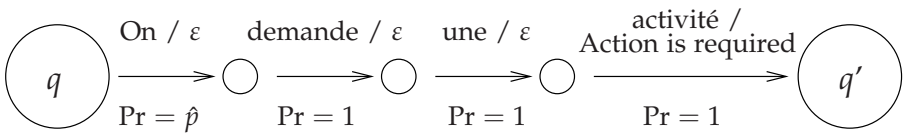


Figura 3.14: Combinación óptima de diversos modelos basados en segmentos. El valor del peso es $\hat{p} = \prod_{m=1}^{M} P_m^{\hat{\lambda}_m}$ cuyo logaritmo es $\log \hat{p} = \sum_{m=1}^{M} \hat{\lambda}_m \log P_m$

3.6. Resumen del capítulo

Este capítulo ha descrito la metodología GIATI para el aprendizaje de transductores estocásticos de estados finitos a través de técnicas basadas en modelado del lenguaje para su empleo en traducción automática estadística. Dicha descripción incluye además su implementación por medio de modelos de n-gramas, cuyo uso en modelado de lenguaje se encuentra muy extendido.

En este sentido, las aportaciones de esta tesis a dicha metodología y a la explotación de los modelos de traducción derivados de ella son las siguientes:

- Un algoritmo de filtrado de modelos mediante una partición de datos. En la dirección de otros sistemas de traducción automática estadística, el modelo estimado se filtra para su aplicación sobre un corpus dado. Este procedimiento se aplica directamente sobre el modelo de n-gramas, generando un transductor más pequeño compatible con dicho corpus.
- La propuesta de GIATI sobre los transductores basados en segmentos. Esta aproximación se ha abordado desde dos perspectivas diferentes, donde, en una de ellas se hace uso de alineamientos estadísticos, y en la otra, de modelos de traducción ya entrenados, basados en segmentos.

- Estos modelos han causado la adecuación de los algoritmos de búsqueda adaptando las estrategias de análisis a sus características inherentes, dando lugar a búsquedas basadas en el modelo de lenguaje subyacente.
- Dicha estrategia de búsqueda nos ha permitido trasladar, de una manera completamente equivalente, el suavizado de los modelos de n-gramas al contexto de los transductores GIATI con propósitos de traducción. Esta equivalencia se implementa de un modo muy sencillo mediante una reinterpretación de las aristas de *backoff* durante la exploración, junto al mantenimiento de un conjunto dinámico de estados prohibidos. En un contexto de traducción, este mecanismo de gestión del suavizado permite una exploración completa del espacio de búsqueda, es decir, encuentra todas las cadenas con las que se relaciona una entrada dada.

Por último, este capítulo ha expuesto también un marco de trabajo log-lineal para la combinación de transductores GIATI con otros modelos estadísticos.

Capítulo 4

Resultados experimentales con GIATI

El rendimiento de esta metodología de estados finitos, aplicada sobre diversas tareas de traducción, se ha evaluado mediante tres medidas de calidad. En algunos casos, en aras de analizar el coste computacional de los prototipos de traducción, se emplean una serie de medidas de eficiencia que permiten obtener una estimación de dicho coste, tanto temporal como espacialmente.

4.1. Medidas de análisis de prestaciones

Desde su aparición como medida de calidad en traducción, el indicador BLEU [PRWZ02] (*BiLingual Evaluation Understudy*) se ha afianzado en el terreno de la evaluación automática como la medida más estándar en el sector. Sin embargo, posteriormente se ha descubierto que su factor de correlación con evaluaciones subjetivas, razón a la que le debe en gran medida su éxito, no es tan elevado como se había pensado en una primera instancia [CBOK06]. A pesar de ello, sigue siendo la medida mayormente utilizada en la literatura. Un indicador muy parecido a BLEU, surgido paralelamente, es NIST [Dod02].

Por otro lado, la tasa WER (*Word Error Rate*) es una medida popular en reconocimiento del habla, aplicada también en traducción automática [OTN99].

Aunque no es tan habitual en traducción, existen ciertos trabajos que demuestran una correlación mayor que BLEU sobre valoraciones humanas [PFS07].

Por último, como evolución de la tasa WER en traducción automática, tras la aparición de la tasa PER (*Position independent Error Rate*) [OTN99], basada asimismo en el concepto de distancia de edición, ha surgido recientemente una medida nueva, más intuitiva, y que rápidamente ha logrado hacerse con un espacio en el área de la evaluación automática en traducción. Su nombre es *Translation Edit Rate* (TER), y su éxito se debe a que modela las ediciones básicas de post-proceso que realizan los traductores profesionales. Estadísticamente, está considerada como una medida cuya correlación con el resultado de una o más evaluaciones subjetivas es también elevada [SDS+06].

Sin embargo, cada día se abren paso un gran número de alternativas a tener en cuenta, entre las que habría que destacar al indicador METEOR [LA07]. Otras medidas que existen en la literatura son, por ejemplo, GTM [MGT03], ParaEval [ZLH06], así como diversas aproximaciones sintácticas [AGGM06], semánticas [GM08], o que combinan estadísticamente varias de ellas [LG07].

Por todos los motivos expuestos, el rendimiento de los modelos de traducción de esta tesis se ha evaluado por medio de estas tres medidas de calidad:

WER : La tasa WER calcula el mínimo número de ediciones (sustituciones, inserciones o borrados) que son necesarias para convertir la hipótesis del sistema en la frase de referencia proporcionada. Esta medida ha sido tan ampliamente utilizada en reconocimiento del habla como criticada en ocasiones [WAC03, CG96]. Una de las críticas más populares que recibe la tasa WER en el área de Traducción Automática reside en la dependencia extrema que se tiene sobre la única referencia de que se dispone para calcular la distancia de edición, cuando suele ser bastante común en lenguaje natural que una misma frase pueda expresarse de muy diversas formas, todas ellas semánticamente equivalentes entre sí. Por su propia naturaleza, la tasa WER es una medida muy pesimista. Una posible solución consiste en extender la tasa WER hacia una medida de error promedio respecto de un conjunto de múltiples referencias.

A este indicador se le conoce como *Multiple Word Error Rate* [NOLN00] y aunque resulta más fiable que la tasa WER, en muchas ocasiones no es posible su aplicación debido a la indisponibilidad de multirreferencias.

BLEU : Este indicador calcula la precisión de los unigramas, bigramas, trigramas, y tetragramas que aparecen en las hipótesis de traducción, respecto de los n-gramas del mismo orden existentes en una o más referencias, con una penalización para las frases de longitud excesivamente corta. A diferencia de la tasa WER, BLEU es una medida de acierto, no de error.

TER : TER calcula, al igual que WER, el número mínimo de ediciones necesarias para convertir una hipótesis en una traducción de referencia. Difieren en el conjunto de operaciones de edición que están permitidas. Además de las incluidas para el cálculo del WER, se añade una operación más, a nivel de segmento, que introduce el desplazamiento de varias palabras contiguas a la vez como una posibilidad más de edición.

Si bien la herramienta empleada para el cálculo del WER es una implementación interna de nuestro grupo de investigación, con respecto a BLEU y TER, las aplicaciones utilizadas son `multi-bleu.perl`[1] y `tercom`[2], respectivamente.

En los casos en los que se desea evaluar el coste computacional de los sistemas de traducción desarrollados, se utilizan dos medidas de eficiencia. Por un lado, la productividad, y por otro, el vocabulario bilingüe empleado:

- La productividad contempla el rendimiento de los sistemas en cuanto a su tasa o velocidad de traducción. Se calcula como la división entre el número de palabras de entrada a traducir y el tiempo requerido para su traducción, sin tener en cuenta los tiempos de carga de los modelos. Por eso, la medida de este factor se expresará en palabras por segundo.
- Los requerimientos espaciales de cada aproximación se pueden estimar por medio del tamaño del vocabulario de símbolos extendidos (o de pares bilingües) que se utiliza en las distintas técnicas de modelado.

[1] Integrada en un *toolkit* de traducción (http://www.statmt.org/moses)
[2] Versión 0.7.2 (http://www.cs.umd.edu/∼snover/tercom)

4.2. Corpus

Esta metodología se ha aplicado sobre diversas tareas de traducción, lo cual ha supuesto una cierta variedad en cuanto a los pares de idiomas usados.

4.2.1. EuroParl

El corpus del EuroParl [Koe05] es el corpus de referencia del *WorkShop on Machine Translation* de la *Association for Computational Linguistics*, en cuyas ediciones anuales se han ido definiendo una serie de tareas compartidas con el objetivo de realizar concursos de traducción de manera que los resultados obtenidos por los distintos participantes pudieran ser comparables entre sí. Con el tiempo, este corpus se ha convertido en un referente a nivel mundial con el que cada miembro de la comunidad científica realiza sus experimentos permitiendo una comparación entre las diferentes aproximaciones existentes.

El corpus del EuroParl se construye a partir de las actas del Parlamento Europeo, las cuales se publican en su página web y están disponibles libremente. A causa de su naturaleza, este corpus presenta una gran variabilidad, en parte debido a que las traducciones se realizan por medio de grupos de traductores profesionales. El hecho de que no todas las personas se pongan de acuerdo en sus criterios de traducción implica que una misma frase de entrada pueda traducirse de modos muy diversos a lo largo de todo el corpus.

Ya que las actas no se encuentran completamente disponibles en todos los idiomas oficiales de la Unión Europea, para cada par de idiomas se ha de extraer, por tanto, un subconjunto distinto. Esto implica, de alguna manera, que los corpus sean algo diferentes en función del par de idiomas a modelar.

Durante las ediciones 2006 y 2007 del citado congreso, se propusieron diversas tareas compartidas de traducción, entre las que seleccionamos algunas de las relativas a idiomas tales como el francés, inglés, o español, debido al conocimiento que el autor de esta tesis posee sobre dichas lenguas. En concreto, la elección recayó sobre la versión francés-inglés y español-inglés

de 2006. No obstante, luego se quiso estudiar el efecto del tamaño del corpus, para lo que se decidió que bastaba con añadir el par español-inglés de 2007, de mayores dimensiones. Las tablas 4.1, 4.2 y 4.3 muestran sus características.

Tabla 4.1: Características del corpus francés-inglés del EuroParl 2006

		francés	**inglés**
Entrenamiento	Nº frases	688031	
	Nº palabras	15.6 M	13.8 M
	Vocabulario	80348	61626
Test	Nº frases	2000	
	Nº palabras	66200	57951
	Perplejidad	49.6	71.6

Tabla 4.2: Características del corpus español-inglés del EuroParl 2006

		español	**inglés**
Entrenamiento	Nº frases	730740	
	Nº palabras	15.7 M	15.2 M
	Vocabulario	102216	64070
Test	Nº frases	2000	
	Nº palabras	60332	57951
	Perplejidad	72.3	70.4

Tabla 4.3: Características del corpus español-inglés del EuroParl 2007

		español	**inglés**
Entrenamiento	Nº frases	964791	
	Nº palabras	20.9 M	20.3 M
	Vocabulario	113026	81754
Test	Nº frases	2000	
	Nº palabras	60243	58059
	Perplejidad	71.2	68.3

En www.statmt.org/matrix hay resultados de referencia para este corpus.

4.2.2. i3media

i3media[3] es un ambicioso proyecto audiovisual cuyos contenidos en materia de traducción se centran exclusivamente en la dirección español→catalán. El corpus de i3media se ha obtenido a partir de la publicación electrónica en Internet de "El Periódico de Cataluña" (http://www.elperiodico.es) por medio de un proceso de recolección automática de textos paralelos [TdVF+01]. Este periódico de información general se publica diariamente en edición bilingüe, en estos dos idiomas. El tipo de lenguaje que se emplea se ubica dentro de un contexto periodístico, incluyendo subsecciones tan habituales como, por ejemplo, editorial, política, deportes, programación televisiva, etc. Las características de este corpus se pueden ver en la tabla 4.4.

Tabla 4.4: Características del corpus i3media

		español	**catalán**
Entrenamiento	Nº frases	805516	
	Nº palabras	14.2 M	14.8 M
	Vocabulario	167670	165512
Test	Nº frases	6000	
	Nº palabras	104899	109114
	Perplejidad	107.3	91.6

Respecto a los resultados de referencia, [Tom03] muestra cifras similares a las de esta tesis aunque las particiones empleadas no son del todo equivalentes. Experimentos internos con los mismos conjuntos de datos sitúan nuestra aproximación basada en transductores finitos a la cabeza de la clasificación.

4.2.3. Xerox

El texto multilingüe original que se utilizó para construir uno de los corpus a utilizar en la fase de experimentación fue proporcionado por Xerox

[3] http://www.i3media.org

Research Center Europe [ELVL04]. El texto contiene múltiples traducciones de diversos manuales de utilización de impresoras Xerox. Los manuales se escribieron inicialmente en inglés y se tradujeron al español, alemán, y francés. El corpus se procesó para la corrección de posibles errores, obteniéndose a su vez diferentes versiones del mismo, según el grado de simplificación logrado. Las características de estos corpus se pueden ver en las tablas 4.5, 4.6 y 4.7.

Tabla 4.5: Características del corpus francés-inglés de Xerox

		francés	**inglés**
Entrenamiento	Nº frases	52844	
	Nº palabras	696 K	633 K
	Vocabulario	9898	7787
Test	Nº frases	984	
	Nº palabras	11858	11152
	Perplejidad	57.3	78.1

Tabla 4.6: Características del corpus español-inglés de Xerox

		español	**inglés**
Entrenamiento	Nº frases	55761	
	Nº palabras	753 K	665 K
	Vocabulario	11042	7953
Test	Nº frases	1125	
	Nº palabras	10106	8370
	Perplejidad	35.3	51.1

Tabla 4.7: Características del corpus alemán-inglés de Xerox

		alemán	**inglés**
Entrenamiento	Nº frases	49376	
	Nº palabras	535 K	588 K
	Vocabulario	19284	7671
Test	Nº frases	996	
	Nº palabras	11692	12322
	Perplejidad	121.3	54.6

Como referencia, los resultados que presentamos a continuación en la sección 4.3 mejoran ostensiblemente los citados en [BBC$^+$09] para este corpus, donde se comparan diversos sistemas.

4.3. Resultados

Se ha diseñado un conjunto de experimentos relativos a los parámetros estudiados tanto en el entrenamiento de transductores estocásticos de estados finitos como en su utilización durante la correspondiente etapa de búsqueda. Para ello, se han empleado los corpus que se han descrito en la sección 4.2.

4.3.1. Transductores basados en palabras vs. segmentos

Según se ha visto en el Capítulo 3, los alineamientos a un nivel de palabra se pueden usar heurísticamente para generar el corpus de cadenas extendidas a partir del que inferir los correspondientes transductores de estados finitos.

Dichas relaciones se estiman estadísticamente como una función uno-a-muchos producto de la aplicación de la herramienta pública GIZA++ [ON03]. Este tipo de alineamientos conducen de forma natural a los llamados transductores basados en palabras, dado que la taxonomía de modelos presentada en la sección 3.3, basada en una granularidad a nivel de palabra o segmento, posibilita la clasificación de éstos por medio de la capacidad de integración de las palabras de entrada en los símbolos extendidos del lenguaje bilingüe.

Por el contrario, los así denominados transductores basados en segmentos necesitan correspondencias muchos-a-muchos como modelo de alineamiento, cuya obtención a partir de las relaciones uno-a-muchos derivadas de GIZA se describe en la subsección 3.3.2 bajo el apartado de *aglutinamiento mínimo*. La tabla 4.8 muestra los mejores resultados mediante ambas aproximaciones, tras variar el orden n de los modelos de n-gramas entre los valores de 1 y 5. En el Apéndice A se puede observar de una manera exhaustiva cómo influye este parámetro del modelo en la calidad de los resultados de traducción

a través de todos los corpus y configuraciones experimentales estudiadas. En líneas generales, la conclusión más significativa derivada de este análisis señala que el salto cualitativo se produce a partir de los modelos de orden 2.

Tabla 4.8: Resultados con modelos basados en alineamientos estadísticos. En cada caso particular, se constata el mejor resultado de los obtenidos por medio de diversos modelos de n-gramas, al variar el parámetro n entre 1 y 5.

		Palabras				**Segmentos**			
Corpus	**Idiomas**	n	BLEU	WER	TER	n	BLEU	WER	TER
2006 Europarl	fr→en	4	20.2	64.1	62.2	3	25.1	65.3	61.1
	es→en	3	20.5	63.5	61.8	3	25.5	64.4	60.2
	en→es	2	16.8	67.9	66.4	4	25.3	64.4	60.2
2007	es→en	3	22.1	62.1	60.1	3	26.3	63.2	58.9
	en→es	2	20.1	64.9	62.6	3	26.0	63.8	59.7
Xerox	en→de	5	18.1	71.4	69.9	2	22.6	70.1	68.1
	de→en	3	24.7	63.8	62.3	3	29.7	58.7	55.8
	en→es	5	58.3	32.1	30.3	2	61.0	28.5	26.4
	es→en	5	52.1	33.8	32.7	2	56.5	29.9	27.6
	en→fr	4	27.9	62.6	60.9	3	33.5	58.1	55.4
	fr→en	3	29.4	56.8	55.3	3	31.7	57.7	54.8
i3media	es→ca	2	79.9	13.4	13.0	3	80.5	12.2	11.8

A partir de los resultados de traducción que se presentan en la tabla 4.8, y siendo que ambas técnicas emplean los mismos alineamientos de entrada, se puede concluir que los transductores de estados finitos basados en segmentos superan claramente a los modelos estrictamente basados en palabras. Los modelos basados en segmentos alcanzan una mejora relativa en torno al 19.5 % de media en lo que respecta al indicador de precisión BLEU para todos los corpus y sentidos de traducción que se han evaluado experimentalmente.

4.3.2. Interpretación de los pesos de *backoff* en el suavizado

Las aristas de *backoff* se pueden interpretar de dos maneras diferentes, asociándose su uso en el contexto del modelo original, es decir, el autómata, o bien en el del transductor. En ambos casos, su significado es el de una

transición de fallo cuya utilización dependerá del tipo de análisis a realizar. Por ejemplo, en el caso del transductor, los símbolos de las cadenas a analizar pertenecen a su alfabeto de entrada Σ, mientras que, en el caso del autómata, deben pertenecer al único alfabeto del que consta, Γ, de naturaleza bilingüe.

Sin embargo, dado que en traducción automática la única información disponible en la fase de búsqueda es la de las cadenas a traducir, pertenecientes a $\Sigma^\star$, el uso de las aristas de *backoff* se ha aplicado históricamente en el contexto del transductor, permitiendo su acceso sólo en caso de necesidad, y siempre que no exista ninguna otra arista de éxito desde ese mismo estado.

En esta línea, nuestra propuesta en esta tesis es la de simular el alfabeto Γ a partir de las frases de entrada en $\Sigma^\star$ para una interpretación de los pesos de *backoff* desde el punto de vista del modelo que lo introdujo como parámetro, es decir, el modelo de n-gramas o su expresión en forma de autómata finito.

Tal y como hemos visto en la sección 3.4.2, dicha interpretación se implementa de una manera sencilla considerando a las aristas de *backoff* como transiciones ε/ε, limitando la exploración por medio de un conjunto dinámico de estados prohibidos que garantice la jerarquía del modelo de n-gramas. Esto se plantea como un *backoff* condicional donde la exploración de niveles inferiores depende de la que previamente se ha realizado en capas más altas.

Los resultados con este algoritmo de suavizado se muestran en la tabla 4.9, mediante los que se confirman de nuevo las conclusiones de la sección 4.3.1 respecto a la superioridad de los transductores GIATI basados en segmentos.

De la comparación entre sí de las tablas 4.8 y 4.9 se deduce que la propuesta de reinterpretación de los pesos de *backoff* en el suavizado es satisfactoria. La repercusión en este caso no es tan notable como la aportación de los transductores basados en segmentos, aunque cabe destacar que los modelos evaluados en cada tarea de traducción son los mismos en cada sección, lo que cambia es la utilización de los arcos de *backoff* durante la fase de búsqueda. Los resultados mejoran en todos los corpus, incrementando la tasa de BLEU una media de 1.5 puntos aproximadamente, lo que permite verificar en la práctica una máxima en modelado estadístico, la de que no sólo importa dis-

Tabla 4.9: Resultados con modelos basados en alineamientos estadísticos cuyas transiciones de *backoff* se reinterpretan como un suavizado condicional. En cada caso particular, se constata de nuevo el mejor resultado obtenido mediante diversos modelos de n-gramas, al variar el parámetro n entre 1 y 5.

		Palabras				**Segmentos**			
Corpus	**Idiomas**	n	BLEU	WER	TER	n	BLEU	WER	TER
2006 Europarl	fr→en	5	21.2	63.0	60.9	4	27.0	63.2	58.9
	es→en	5	21.4	62.4	60.4	4	27.0	62.8	58.4
	en→es	2	16.8	67.6	66.1	4	26.2	63.2	59.1
2007	es→en	3	23.0	61.2	59.1	4	27.9	61.6	57.2
	en→es	5	20.4	64.2	62.5	3	26.8	62.6	58.4
Xerox	en→de	4	18.6	70.7	69.2	4	23.0	69.2	67.5
	de→en	3	25.4	63.2	61.6	3	30.2	58.2	55.3
	en→es	4	58.3	31.5	29.8	3	62.3	27.5	25.3
	es→en	5	52.5	33.3	32.2	4	58.9	28.0	25.8
	en→fr	4	28.3	62.4	60.5	3	34.4	57.7	54.9
	fr→en	4	30.2	56.4	54.7	4	32.6	57.6	54.7
i3media	es→ca	4	82.6	10.8	10.5	5	83.5	10.2	9.9

poner de un modelo adecuado, sino también del uso que se hace del mismo.

En adelante, la búsqueda se suaviza mediante dicho *backoff* condicional cuya eficacia sobre la exploración de modelos GIATI ha quedado demostrada. En cualquier caso, el Apéndice A muestra los resultados con ambos métodos.

4.3.3. Transductores basados en otros sistemas de traducción

Tal y como se ha descrito en la sección 3.3.2, los pares de segmentos de un modelo de traducción se pueden utilizar en GIATI como símbolos extendidos con la ayuda de un sistema de traducción automática de carácter monótono. En este caso, las segmentaciones asociadas se aproximaron mediante el sistema Pharaoh [Koe04], cuyo modelo de traducción está basado en segmentos.

Por otro lado, dado que el corpus del EuroParl resulta el más interesante, por tamaño y dificultad, esta técnica se aplica únicamente sobre este corpus.

La tabla 4.10 muestra las calidades de los modelos GIATI basados en Pharaoh.

Tabla 4.10: Resultados de traducción sobre los corpus del EuroParl logrados por medio de transductores basados en el modelo de traducción de Pharaoh.

Corpus	**Idiomas**	n	BLEU	WER	TER
2006 EuroParl	fr→en	2	28.0	61.9	57.3
	es→en	3	27.6	61.7	57.1
	en→es	2	26.4	62.3	57.7
2007	es→en	4	28.1	59.5	54.9
	en→es	2	25.2	60.8	56.6

Salvo en el último caso donde el BLEU baja (aunque WER y TER también), las cifras indican una mejora relativa respecto a las de la subsección anterior de en torno al 1.9 % respecto a los transductores basados en aglutinamiento. También se aprecian menores n, sugiriendo unos modelos más optimizados. De hecho, si consultamos el Apéndice A para ver el efecto de esta variable, resulta asimismo significativo que las diferencias entre el modelo de unigramas y el resto de modelos de orden superior se hayan visto reducidas al mínimo.

Esta aproximación, cuya viabilidad acabamos de presentar, se compara a continuación con el sistema de traducción del que proviene. Debido a que los resultados sobre el corpus del EuroParl parece que siguen una pauta similar, hemos limitado el siguiente experimento sobre la partición francés→inglés, juzgando extrapolables a todos los efectos las conclusiones derivadas de éste. Para ello, hemos utilizado Moses [KHB+07] como herramienta de traducción, debido a que representa la evolución y el estado del arte del sistema Pharaoh.

Para este experimento, se han comparado dos sistemas de traducción automática: uno construido exclusivamente mediante los programas de Moses, que tomaremos como sistema de referencia con el propósito de compararlo con respecto al sistema que se ha desarrollado a través de la transferencia del correspondiente modelo de traducción (entrenado mediante Moses) al marco de trabajo de estados finitos derivado de la aplicación de un algoritmo GIATI.

En cuanto a Moses, la configuración que se ha utilizado para la búsqueda incluye todos los modelos habituales excepto el modelo de reordenamiento, es decir, dos modelos de traducción basados en segmentos (directo e inverso), otros dos modelos de traducción, pero en esta ocasión, basados en palabras, un modelo de lenguaje de salida, un modelo de longitud de los segmentos, y, por último, un modelo para penalizar las hipótesis de longitud muy corta. Por otro lado, reiteramos que GIATI sólo emplea un modelo: el transductor.

Evaluamos también el rendimiento de Moses según los modelos usados. Por un lado, añadimos un modelo de reordenamiento basado en segmentos, y por otro, contemplamos un experimento con sólo un modelo de traducción, el modelo de lenguaje de salida, y el de penalización de las hipótesis cortas. Los resultados de traducción con Moses y GIATI se observan en la tabla 4.11. Respecto a los de Moses, se confirman con los publicados para este corpus[4].

Tabla 4.11: Resultados sobre el corpus francés→inglés del EuroParl 2006 comparando el sistema Moses con el transductor GIATI derivado de éste. Dicho transductor se basa en un modelo de bigramas de segmentos bilingües. Se constata asimismo el número M de modelos usados en cada metodología, siendo que $M = 7$ es la versión de Moses con restricción de monotonicidad; con $M = 8$ se añade además un modelo de reordenamiento de segmentos; $M = 3$ sólo supone un modelo de traducción, de lenguaje, y de penalización.

Sistema	M	BLEU	WER	TER
Moses	3	26.9	61.2	57.1
Moses	7	30.3	58.0	53.6
Moses	8	30.6	57.7	53.3
GIATI	1	29.3	58.9	54.5

El transductor GIATI se ha construido a partir de una restricción monótona sobre Moses, por lo que es con esta versión con la que se debe comparar. Como se puede apreciar, la transferencia hacia estados finitos de modelos de

[4]Véase http://www.statmt.org/matrix

traducción basados en segmentos es bastante eficiente en esta tarea. Más aún si tenemos en cuenta que un transductor es realmente un modelo bastante más sencillo que Moses o que cualquier otra tecnología basada en segmentos. La caída de 1 punto en las medidas de calidad se debe muy probablemente a la reducción en el número de pares de segmentos bilingües contemplados, provocada por la adaptación entre metodologías de las tablas de traducción. De hecho, la selección de segmentos de Moses para su uso mediante GIATI representa un subconjunto que contiene únicamente el 6.8 % de las entradas.

Los resultados confirman la escasa aportación que tiene el reordenamiento en esta tarea, sugiriendo que una aproximación monótona es bastante capaz. El transductor GIATI permite simular el rendimiento monótono de Moses mediante un único modelo, frente a la configuración original de 7 modelos, cuya capacidad se reduce bastante más si se usa un número menor de éstos. Como un dato a tener en cuenta, la utilización de Moses con sólo 3 modelos, lo que permite una comparación en similares condiciones a las de GIATI, presenta una cierta pérdida de calidad de traducción respecto al transductor.

En cuanto a la eficiencia temporal de los sistemas de traducción usados, expresada en términos de su velocidad e indicada en palabras por segundo, hay que hacer constar que mientras los experimentos relativos a Moses presentan unos ratios que oscilan entre 18 y 27, según el número de modelos, la velocidad de procesamiento del transductor GIATI está en torno a las 200. De este modo, la diferencia de calidad entre los resultados a través de GIATI y los que se obtienen a partir de una configuración estándar mediante Moses resulta despreciable en la práctica al observar los beneficios computacionales de un marco de trabajo basado en modelos de estados finitos respecto de sistemas como Moses, y sobre todo su aplicación en entornos de tiempo real.

Por otro lado, el transductor GIATI basado en los segmentos de Moses supera al que está basado en los de Pharaoh, lo que también era de esperar. En cualquier caso, estos transductores basados en modelos de traducción tienen limitado su rendimiento al de las herramientas de las que dependen, sin embargo, esto no ha impedido lograr los mejores resultados de esta tarea.

4.3.4. Aceleración del proceso de búsqueda

Se ha diseñado un conjunto de experimentos para evaluar el impacto de las técnicas propuestas sobre aceleración de la búsqueda a través de transductores de estados finitos GIATI basados en n-gramas de segmentos bilingües.

Con este objetivo, el corpus utilizado en esta fase experimental ha sido de nuevo el EuroParl, a través de la partición francés→inglés del año 2006, cuyas tasas de calidad para cierto transductor GIATI basado en segmentos, habiendo explorado el espacio de búsqueda sin haber usado técnicas de poda, se pueden consultar en la primera fila de la tabla 4.10, con un BLEU de 28.0.

En aras de una mayor comprensión, recordamos que la tasa de eficiencia empleada es el rendimiento del sistema, expresado en palabras por segundo. Los resultados de traducción junto con los de eficiencia computacional para estrategias de búsqueda basadas en palabras o segmentos, en función de la dimensión estática del haz, se pueden comparar en la gráfica de la figura 4.1. Los relativos a una exploración con haz dinámico se reflejan en la gráfica 4.2.

De la observación de las figuras 4.1 y 4.2, se puede deducir que una estrategia basada en palabras tiene en cuenta iterativamente un alto porcentaje de estados inútiles, por lo que resulta necesario incrementar el tamaño del haz para contemplar también los caminos de éxito a lo largo de la búsqueda. El precio que se paga por tener que considerar una lista de estados tan larga en cada iteración del algoritmo es en términos de requerimientos temporales.

Sin embargo, una estrategia basada en segmentos sólo almacena aquellos estados que se han alcanzado con éxito tras una compatibilidad completa a nivel de segmento. Por lo tanto, se requiere más tiempo para procesar cada estado de forma individual, pero debido a que el número de estados a tener en cuenta en cada iteración es menor, el rendimiento global es muy superior.

Por otro lado, el modelo completo aprendido por medio del conjunto de entrenamiento presenta un total de aproximadamente 6 millones de eventos, entre unigramas y bigramas. La aplicación de la poda de n-gramas descrita en la sección 3.2.1, utilizando una ventana de tamaño creciente, permite la

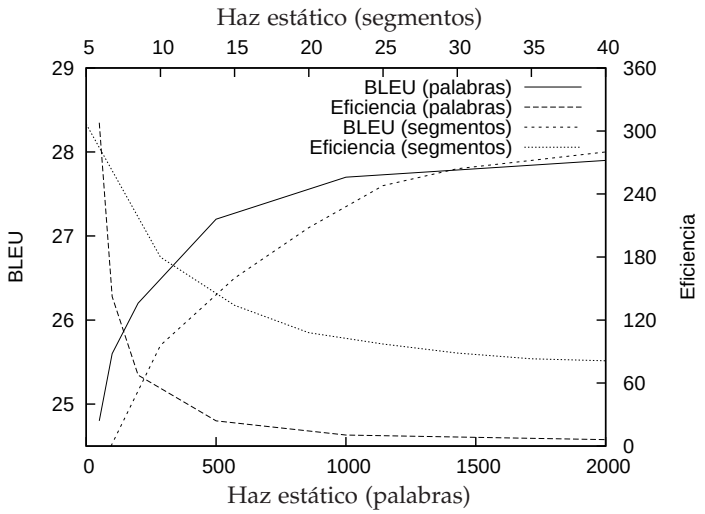


Figura 4.1: Haz estático sobre búsquedas basadas en palabras o segmentos. Se reporta la calidad y eficiencia de un transductor basado en un modelo de bigramas de segmentos bilingües sobre la tarea francés→inglés del EuroParl.

reducción de dicho número a apenas unos 200000 eventos que son compatibles con el corpus de test. Estos n-gramas, transformados como transiciones de un transductor GIATI, suponen un descenso en el número de aristas del modelo en torno a un 97 % (desde casi unos 20 millones hasta sólo 500000). Como consecuencia, al disminuir el tamaño del modelo, el tiempo de búsqueda también decrece. Las figuras 4.3 y 4.4 muestran el efecto de la poda de n-gramas sobre el tamaño de los transductores, y consecuentemente, sobre su eficiencia por medio de una estrategia de búsqueda basada en segmentos.

4.3.5. Integración en un entorno log-lineal

El marco de trabajo log-lineal que planteamos en esta sección se basa en la integración de un conjunto de $M = 3$ transductores basados en segmentos. Para ello, el escenario que se ha diseñado consta de un transductor GIATI, como modelo principal, y de la colaboración de dos modelos de traducción

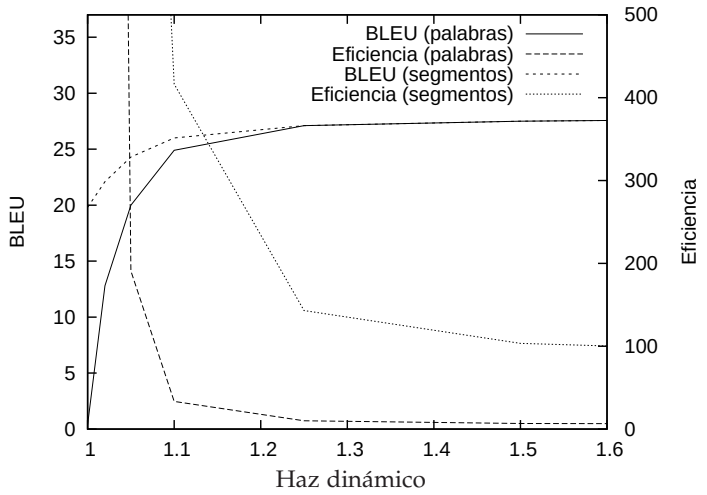


Figura 4.2: Haz dinámico sobre búsquedas basadas en palabras o segmentos. Se reporta la calidad y eficiencia de un transductor basado en un modelo de bigramas de segmentos bilingües sobre la tarea francés→inglés del EuroParl.

(sendos modelos directo e inverso), que se evalúan sobre pares de segmentos en función de las relaciones que hay entre sus palabras de entrada y de salida.

La sección 3.3.2 describe dos opciones para construir transductores GIATI basados en segmentos. Sin embargo, los transductores basados en otros sistemas de traducción se consideran ya producto de la integración de modelos de los sistemas de los que proceden. Por lo tanto, esta elección carece de interés en este contexto ya que dichos sistemas suelen incluir en su configuración diversos modelos de traducción, lo que hará inútil la combinación propuesta. En cambio, la aproximación mediante aglutinamiento mínimo es adecuada en este escenario, bajo el que a través de la combinación con otros modelos, podrían alcanzarse las cotas de los transductores basados en Pharaoh o Moses sin necesidad de depender de dichas herramientas durante el entrenamiento.

Las probabilidades a nivel de segmento de los modelos de traducción se calculan bajo una aproximación basada en el modelo 1 de IBM [BPPM93],

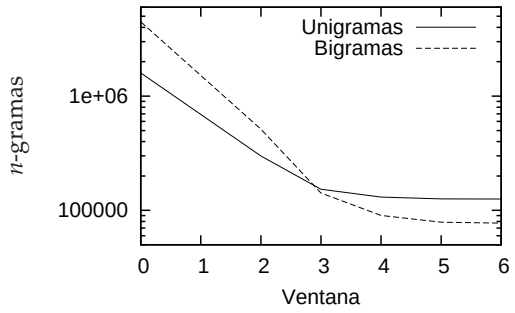


Figura 4.3: Efecto de la ventana de muestreo sobre la poda de n-gramas de un modelo basado en segmentos para el corpus francés→inglés del EuroParl.

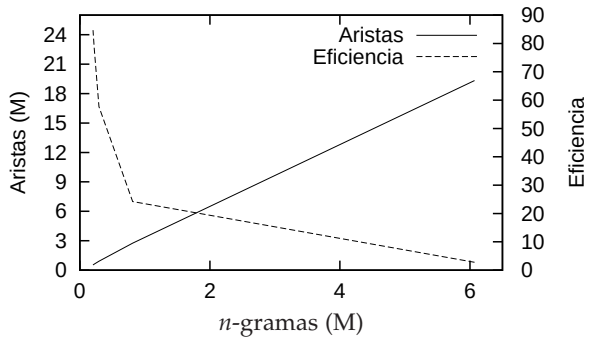


Figura 4.4: Efecto del nº n-gramas sobre el tamaño y la eficiencia de un transductor basado en segmentos para la tarea francés→inglés del EuroParl.

utilizando un diccionario estocástico de palabras para ponderar las relaciones internas de traducción, normalizadas por medio de una distribución de probabilidad de Poisson, aplicada sobre las longitudes de los segmentos [SH07]:

$$\Pr(l_s|l_t) = \frac{e^{-l_t r}(l_t r)^{l_s}}{l_s!}$$

$$\Pr(\mathbf{s}_j^{j'}|\mathbf{t}_i^{i'}) = \Pr(l_{\mathbf{s}_j^{j'}}|l_{\mathbf{t}_i^{i'}}) \prod_{j''=j}^{j'} \sum_{i''=i}^{i'} \Pr(\mathbf{s}_{j''}|\mathbf{t}_{i''})$$

donde l_s y l_t son las correspondientes longitudes de segmento de entrada y

de salida, y donde $r = \bar{l}_s / \bar{l}_t$ es el cociente entre sus valores medios globales. Estas expresiones se refieren a un modelo de traducción inverso de $\Pr(\mathbf{s} \mid \mathbf{t})$. Para un modelo directo de $\Pr(\mathbf{t} \mid \mathbf{s})$, las ecuaciones que se usan son similares.

Los diccionarios estocásticos basados en palabras se infieren también por medio del paquete GIZA++, a través del mismo proceso en el que, como subproducto, se obtienen los alineamientos necesarios para las segmentaciones.

Una vez estimados todos los modelos basados en segmentos, se construye un transductor extendido siguiendo el método expuesto en la sección 3.5.2. Dicho transductor contempla todos los modelos a la vez mediante el aumento del número de imágenes de la función asociada a las transiciones del modelo.

Este escenario se ha aplicado sobre el corpus francés→inglés del EuroParl, a partir del modelo basado en aglutinamiento mínimo de mayor rendimiento, es decir, el reflejado en la tabla 4.9 para transductores basados en segmentos.

Se realizó una optimización de los parámetros λ sobre una partición disjunta de desarrollo, de características similares a la de test, usando la tasa BLEU como función de maximización. Los pesos $\hat{\lambda}$ óptimos encontrados fueron de 75.1 % para el modelo GIATI, de 19.0 % para el modelo directo de traducción, y de 12.5 % para el modelo inverso de traducción. En la tabla 4.12 se muestran los resultados de dicha combinación log-lineal de transductores así como los correspondientes a una utilización independiente de los mismos.

Tabla 4.12: Rendimiento individual y colectivo de la combinación de transductores basados en segmentos sobre el corpus francés→inglés del EuroParl.

pesos λ					
GIATI	directo	inverso	BLEU	WER	TER
100 %	0	0	27.0	63.2	58.9
0	100 %	0	5.8	68.4	67.0
0	0	100 %	4.6	104.8	103.1
75.1 %	19.0 %	12.5 %	28.4	60.2	55.9

Tal y como se esperaba, los modelos de traducción individuales son mo-

delos muy pobres aisladamente que requieren su integración en un marco de traducción más general. Es destacable, sin embargo, que el modelo directo refleje un impacto menor en los indicadores de error que el modelo inverso. En cualquier caso, la combinación de éstos con el modelo GIATI en un marco de modelado log-lineal permite superar las espectativas iniciales de calidad, incrementando el rendimiento a través de una simple operación de *rescoring*.

Además, el modelo integrado resultante es incluso un 6.1 % más pequeño que el transductor GIATI original ya que el modelo combinado puede adquirir ventaja de las probabilidades locales de traducción que valen 0, aplicando una interesante técnica de poda cuyo cometido es eliminar del modelo todas las transiciones relativas a los correspondientes pares de segmentos bilingües.

El aumento de las prestaciones no está garantizado en ningún caso aunque intuitivamente podemos decir que esta técnica tendrá tanta repercusión como pobreza estadística posea el transductor GIATI inferido, es decir, con corpus de alta variabilidad, los parámetros del modelo generalmente se aprenden peor, por lo que la incorporación de otros modelos ayuda a su reestimación. Tal es el caso de la tarea de traducción del EuroParl, por ejemplo, cuyo efecto sobre la partición estudiada consideramos extrapolable al resto del corpus. No así respecto a las otras dos tareas, Xerox e i3media, cuyo resultado se ha comprobado explícitamente a través de la correspondiente experimentación.

El dominio del corpus de Xerox es de menor amplitud que el del EuroParl, lo que explicaría que la adición de otros modelos bajo el escenario log-lineal no haya afectado la calidad de los transductores GIATI basados en segmentos, con la única excepción del sentido francés→inglés donde sí ha fructificado. Respecto a i3media, el escenario log-lineal propuesto también ha tenido éxito. La tabla 4.13 muestra la eficacia de esta metodología sobre dichas particiones.

Esta idea se explota con éxito en [MBC+06], donde el modelo principal es también un modelo de n-gramas de símbolos extendidos, que se combina mediante sendos modelos léxicos de traducción, similares a los descritos, además de unos modelos de lenguaje y de longitud sobre las frases de salida. Esta tecnología se presenta con resultados competitivos en el estado del arte,

Tabla 4.13: Incremento de rendimiento mediante la combinación log-lineal. Se reportan resultados sobre i3media y la partición francés→inglés de Xerox.

Corpus	**Idiomas**	**Modelo**	BLEU	WER	TER
Xerox	fr→en	GIATI	32.6	57.6	54.7
		GIATI+dir+inv	34.2	52.3	49.2
i3media	es→ca	GIATI	83.5	10.2	9.9
		GIATI+dir+inv	83.9	9.9	9.5

lo que permite un horizonte de esperanza a nuestro marco de estados finitos, en tanto en cuanto incorpore el resto de modelos bajo el escenario log-lineal.

4.4. Análisis de errores

Para comprender mejor el significado real de los indicadores de calidad, hemos realizado un análisis de las traducciones que produce el modelo GIATI cuyas medidas de evaluación automática presentan las mejores perspectivas sobre los resultados con la partición francés→inglés del corpus del EuroParl.

En la tabla 4.14, un resultado de traducción cuyo BLEU aislado es de 19.3, es decir, 10 puntos por debajo del resultado global del corpus de evaluación, por lo que este ejemplo puede considerarse como uno de los resultados *malos*, cuya repercusión hace bajar la media de calidad que está asociada a esta tarea.

A simple vista, la hipótesis no es tan mala como las cifras pueden indicar. A grandes rasgos, se observa que el sistema de traducción tiende a literalizar la traducción que genera respecto de la frase original que tiene que traducir, lo que se representa mediante el formato cursiva sobre el texto de dicha tabla, mientras que la referencia presenta una versión más compacta y casi literaria, más cercana a una aplicación de corpus comparable, que no paralelo [Bor02].

Los problemas adelantados en la sección 1.3 se observan de forma patente, donde la dependencia con una única referencia penaliza otras posibilidades. Fenómenos como la sinonimia se castigan aunque no tendría que ser así.

Tabla 4.14: Análisis de las diferencias entre la hipótesis y su única referencia. La frase original en francés es la marcada como **s**, su expresión en inglés es **r**, y la hipótesis de traducción mediante el transductor GIATI se indica como **t**. Las equivalencias internas se destacan a través de distintos formatos de texto.

s	nous savons très bien que les traités actuels ne suffisent pas *et qu ' il sera nécessaire à l ' avenir* *de développer une structure plus efficace et différente pour l ' union ,* une structure plus constitutionnelle qui indique clairement *quelles sont les compétences des états membres* *et quelles sont les compétences de l ' union .*
t	we know very well that the current treaties are not enough *and that it will be necessary in future* *to develop a structure more effective and different for the union ,* a structure more constitutional which clearly indicates *what are the competences of the member states* *and what are the competences of the union .*
r	we know all too well that the present treaties are inadequate *and that the union will need a better and different structure in future ,* a more constitutional structure which clearly distinguishes *the powers of the member states and those of the union .*

Ejemplos como **all too/very**, **present/current**, o **inadequate/not enough** pueden considerarse grupos equivalentes dentro del contexto de esta frase. Por otro lado, detectamos ciertos problemas de reordenamiento sin resolver, cuya solución correcta tampoco encuentra el modelo de distorsión de Moses, representando por tanto uno de los desafíos pendientes en el estado del arte.

En la tabla 4.15, un resultado de traducción cuyo BLEU aislado es de 60.9, es decir, 30 puntos por encima del resultado global del corpus de evaluación, por lo que este ejemplo se puede considerar como un resultado de los *buenos*, cuya repercusión permite aumentar la calidad media asociada a este corpus.

Nuevamente, vemos que algunas de las diferencias no son tales errores:

Tabla 4.15: Análisis de las diferencias entre la hipótesis y su única referencia. Las diferencias en traducción se resaltan mediante el formato de texto negrita.

s	monsieur le président , je dirais à m. nogueira que , évidemment , dans le modèle que nous avons défini , la politique monétaire est du ressort de la banque centrale européenne et cette dernière prend des décisions en fonction des critères établis dans le traité .
t	mr president , i would say to mr nogueira that **, of course** , in the model that we have defined , monetary policy is the responsibility of the **european** central bank and **this last** takes **decisions** **according to** the criteria in the treaty .
r	mr president , i would say to mr nogueira that **clearly** , in the model that we have defined , monetary policy is the responsibility of the central bank and **it is the european central bank that** takes **action** **in line with** the criteria **established** in the treaty .

clearly por **, of course**, **action** por **decisions** o **in line with** por **according to**. El único error que se puede considerar como tal es la omisión de **established** como traducción de **établis**, a pesar de que semánticamente no es necesario. En otro orden de discusión está el matiz sobre cómo traducir **cette dernière**, cuya referencia introduce una mención explícita al término al que se refiere, lo que hoy en día todavía está muy lejos de ser resuelto de forma automática.

Hacia el otro extremo, en la tabla 4.16, varios resultados de traducción entre las frases peor puntuadas. Veamos si efectivamente muestran tales errores.

En el primer ejemplo, la omisión de **have** es un error, aunque no es vital. El resto de diferencias se refieren al hecho ya comentado de la preferencia del sistema de traducción automática por las hipótesis literales sobre la entrada. Con el segundo y tercer ejemplo, nos topamos con más de lo mismo agravado mediante la dependencia sobre la única referencia de traducción disponible.

Tabla 4.16: Análisis de las diferencias entre diversas hipótesis y su referencia. La frase $\mathbf{t}_1$ refleja un WER = TER = 87.5; la frase $\mathbf{t}_2$ muestra un WER = 100, TER = 87.5; finalmente, la frase $\mathbf{t}_3$ indica un valor de WER = 100, TER = 94.7.

$\mathbf{s}_1$	c ' est donc à regret que j ' ai voté contre cette proposition .
$\mathbf{t}_1$	it is therefore regret that i voted against this proposal .
$\mathbf{r}_1$	regretfully therefore , i have voted against .
$\mathbf{s}_2$	on doit pouvoir y lire ce qui a été dit par mme van der laan - et nous sommes tous d ' accord là-dessus .
$\mathbf{t}_2$	we must be able to be read what has been said by mrs van der laan - and we are all in agreement on that .
$\mathbf{r}_2$	the points mrs van der laan made earlier should be included , and i am pleased to say that we agree about that .
$\mathbf{s}_3$	et voilà que je dois entendre m. gollnisch dire que son groupe parlementaire a été à la pointe du combat dans la crise de l ' esb .
$\mathbf{t}_3$	and that is that i must hear mr gollnisch say that its group has been at the forefront of the campaign in the bse crisis .
$\mathbf{r}_3$	then mr gollnisch turns round and tells this house that his group has been spearheading the bse debate .

No es difícil encontrar referencias alternativas con una menor penalización. Por ejemplo, la frase *it is therefore* **regretful** *that i* **have** *voted against this proposal* es una traducción correcta de $\mathbf{s}_1$, cuya divergencia de $\mathbf{t}_1$ es de sólo dos errores.

En general, la impresión es la de que el sistema de traducción evaluado, derivado de un marco basado en transductores estocásticos de estados finitos, produce una calidad superior a la que sugieren los indicadores automáticos, por lo que estos sistemas parece que pueden ser útiles en un entorno de CAT.

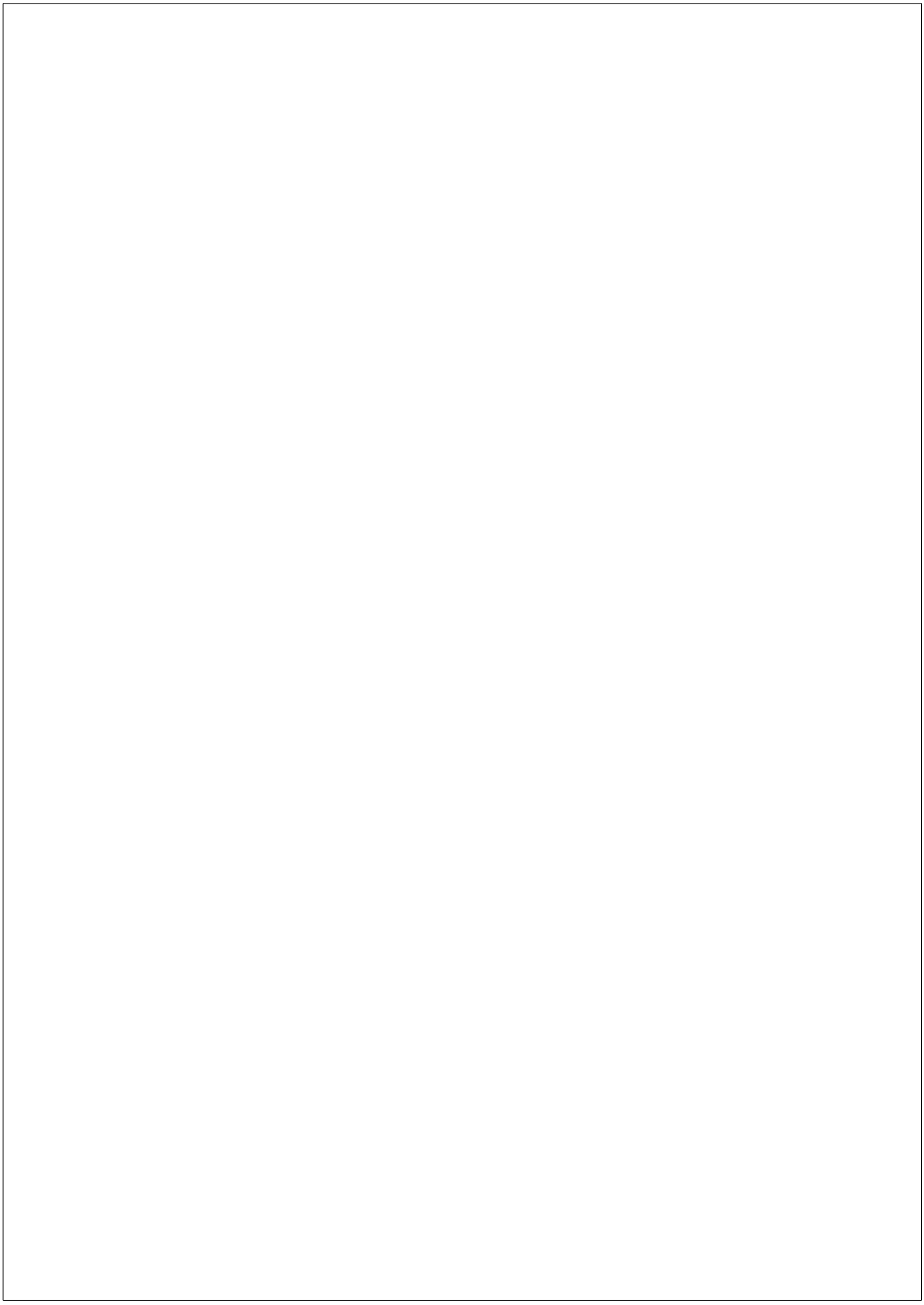
4.5. Resumen del capítulo

Este capítulo ha descrito el desarrollo experimental que se ha realizado en el marco de una implementación de GIATI a través de modelos de n-gramas. Dicha experimentación se ha basado en los desarrollos del capítulo anterior, cuya aplicación en la práctica se ha verificado y contrastado a lo largo de éste.

El capítulo comienza con un análisis de los corpus bilingües empleados y de las medidas de evaluación automática usadas como índice de rendimiento. A continuación, se detallan los resultados obtenidos en el contexto de GIATI:

- Demostración de software escalable mediante tareas como el EuroParl.
- Comparación entre transductores basados en palabras o en segmentos. La propuesta basada en segmentos indica mayor calidad de traducción.
- Comparación de los dos usos de las aristas de *backoff* en el suavizado. La idea basada en aristas ε/ε y prohibición de estados mejora el *baseline*.
- Comparación entre los transductores basados en modelos de traducción y el sistema de traducción automática estado del arte del que proceden. Los transductores aproximan eficientemente los mecanismos de traducción de sistemas que utilizan para su funcionamiento diversos modelos.
- Estudio de estrategias de análisis basadas en palabras o en segmentos para algoritmos de búsqueda con restricciones de poda mediante haz. La eficacia y eficiencia aumentan con búsquedas basadas en segmentos.
- Análisis del algoritmo de filtrado de transductores basado en n-gramas. Estudio de la relación entre eficiencia temporal y el tamaño del modelo.
- Síntesis de la integración log-lineal de varios transductores estocásticos. Una simple operación de *rescoring* permite incrementar las prestaciones.

Finalmente, el capítulo concluye con un análisis de errores más frecuentes que conduce a una valoración bastante optimista de los resultados obtenidos.



Capítulo 5

Morfología en traducción automática

Morfología es el estudio de la manera en la que se construyen las palabras a partir de ciertas unidades más pequeñas llamadas morfemas [Mat92, JM08]. Un morfema se suele definir como la unidad mínima con significado propio en un idioma. Así, por ejemplo, el sustantivo *mesa* se compone de un único morfema (el morfema *mesa*) mientras que su plural *mesas* se compone de dos: el morfema *mesa* y el morfema *-s*.

Tal y como sugiere este último ejemplo, resulta útil distinguir entre dos amplias clases de morfemas: las raíces y sus afijos. Los detalles exactos sobre sus diferencias varían según el idioma, aunque intuitivamente podemos decir que la raíz es el morfema más importante de la palabra, es decir, el que le proporciona su significado principal, mientras que los afijos añaden una serie de significados adicionales de diferente naturaleza.

A su vez, los afijos se dividen en prefijos, sufijos, infijos, y circunfijos. Los prefijos preceden a la raíz, los sufijos la siguen, los circunfijos la rodean por ambos lados, y los infijos se insertan dentro de ella. El término *mesas* al que hacíamos referencia anteriormente es un ejemplo de unión entre raíz y sufijo. En cambio, la palabra *descanso* se compone de la raíz *canso* y del prefijo de negación *des-*. Sin embargo, aunque en castellano no existan buenos ejemplos de circunfijos o de infijos, sí suelen aparecer muy a menudo en otros idiomas.

En alemán, por ejemplo, el participio de algunos verbos se forma añadiendo el morfema *ge-* a la izquierda de la raíz y el morfema *-t* a la derecha de ésta; así, el participio del verbo *sagen* (decir) es *gesagt* (dicho). Los infijos son muy comunes, por ejemplo, en tagalo, idioma de las islas Filipinas. Como muestra de ello, el infijo *um*, que marca el agente de una acción, se inserta en medio de la raíz *hingi* (prestar) para producir la palabra *humingi*.

Al uso de prefijos y sufijos se le suele conocer en lingüística como morfología concatenativa ya que las palabras se componen mediante la concatenación de morfemas. Existen idiomas con una amplia morfología no concatenativa, en la que los morfemas se combinan de una manera más compleja [HS09]. El ejemplo anterior sobre los infijos en tagalo es una muestra de morfología no concatenativa. En árabe, hebreo, y otras lenguas semitas, existe una clase de morfología que también se puede identificar como del tipo no concatenativo.

Como es natural, las palabras pueden estar compuestas de varios afijos. Como ejemplo, la palabra *reescribir* se compone del prefijo *re-*, de la raíz *escrib*, y del sufijo *-ir*. Palabras como *increíblemente* están formadas de una raíz (*cre*) y de tres afijos (*in-*, *-ble*, y *-mente*). Mientras que en español o en inglés los términos tienden a no incluir más de cuatro o cinco afijos, en idiomas como el turco puede ser habitual encontrar palabras con hasta nueve o diez afijos. Idiomas a los que les sucede este fenómeno, es decir, que tienden a encadenar afijos de una manera considerable, se denominan *idiomas aglutinativos* [SB99].

Existen dos grandes familias (parcialmente solapadas) que gobiernan la construcción de palabras a partir de morfemas, conocidas como flexión y derivación. La flexión es la combinación de un morfema de raíz y de un afijo, obteniendo una palabra cuya categoría gramatical no varía con respecto a su raíz, y cuyo significado añadido viene a desempeñar generalmente una función sintáctica, como por ejemplo, de concordancia [Stu01]. En cambio, la derivación produce palabras de una categoría gramatical distinta a la de su raíz correspondiente, frecuentemente con un significado difícil de predecir. Por ejemplo, la flexión verbal en castellano forma el gerundio a través del morfema *-ndo*, mientras que *morosidad* sería una palabra derivada de *moroso*.

5.1. Análisis morfológico mediante estados finitos

No cabe duda de que la morfología es una realidad lingüística que no parece que debiera descartarse de partida en ninguna de las disciplinas del procesamiento del lenguaje natural, incluyendo a la Traducción Automática. Parece evidente que es necesario que los sistemas contemplen que las palabras *vamos* e *iré* deberían ambas traducirse por formas del verbo *go* en inglés.

A pesar de que, en traducción automática estadística, el componente lingüístico que subyace intrínsecamente ha sido generalmente ignorado, en los últimos tiempos ha habido un esfuerzo en materia de investigación, discretamente recompensado, por incorporar cierto conocimiento lingüístico que facilite el proceso de traducción [NN04, DGM06, KH07, PTC06, PGJ+07, PTC08].

Un analizador morfológico consta de, al menos, los siguientes elementos:

- Un léxico, es decir, una lista de raíces y afijos, junto con su información sintáctica básica, por ejemplo, si un cierto morfema es nominal o verbal.
- Un modelo de la formación de palabras a partir del léxico de morfemas. Se establece, de alguna manera, cómo construir palabras válidas en el lenguaje a partir de la combinación de sus morfemas correspondientes.
- Unas reglas de ortografía, que modelan los cambios que suceden dentro de una palabra cuando ciertos morfemas se combinan para construirla.

Existen diferentes modos de modelar toda esta información, aunque una de las más comunes, especialmente si hablamos de morfología concatenativa, es por medio de transductores de estados finitos, que modelen la así llamada en lingüística *morfología de dos niveles*, introducida por vez primera en [Kos83].

La morfología de dos niveles representa a las palabras como una correspondencia entre el nivel léxico, es decir, una simple concatenación de los morfemas de los que están compuestas, y el nivel de la superficie, es decir, su expresión final en forma textual, más visual, y ortográficamente correcta.

Por tanto, el análisis morfológico se implementa mediante un transductor cuyo alfabeto de entrada se corresponde con el nivel superficial de las palabras, mientras que el alfabeto de salida lo hace con el nivel léxico de los morfemas.

Existen diversos entornos de traducción automática, de naturaleza deductiva, en los que se utiliza un análisis morfológico basado en transductores de estados finitos como parte del proceso de traducción [GLN+06, AOCBF+07]. El objetivo consiste entonces en trasladar dicha tecnología lingüística al servicio de la traducción estadística, incorporando morfología en su arquitectura.

Dado que el eje central de esta tesis se basa en una aproximación a la traducción automática estadística por medio de modelos de estados finitos, la integración de dicho conocimiento a priori parece una opción más cómoda de realizar en este tipo de entornos que en otra clase de modelos estadísticos. Esta es otra de las ventajas de los transductores como modelos de traducción.

5.2. Marco estadístico

En todo proceso de reconocimiento de formas, existe una etapa conocida con el nombre de extracción de características que permite describir a los individuos de la población a modelar mediante una codificación computable. Por un lado, una extracción de características geométrica define un objeto **s** como un vector de características en el que cada característica a tener en cuenta se observa sobre **s**, anotando su valor en el campo correspondiente. Por otro lado, una extracción de características sintáctica establece una descripción estructural de **s** mediante un determinado conjunto de primitivas básicas que se interrelacionan por medio de ciertas operaciones gramaticales.

A pesar de ello, la aplicación de esta etapa suele pasar especialmente inadvertida en traducción automática estadística, del mismo modo que también sucede en otros marcos de trabajo relacionados con el lenguaje natural. Debido a que la población se compone de frases en una determinada lengua, y que su transcripción textual representa una descripción estructural de las mismas (mediante la concatenación de símbolos que forman una cadena),

esta codificación en forma de secuencia de palabras se ha venido utilizando tradicionalmente en las tecnologías de lingüística computacional como resultado implícito de un procedimiento oculto de extracción de características.

Sin embargo, nadie ignora el hecho de que la naturaleza lingüística de los idiomas podría explotarse estadísticamente para obtener un mejor modelado. En la literatura, esta aproximación se explora con diversas nomenclaturas, como la de modelos *jerárquicos* [NN04] o la de modelos *factoriales* [KH07]. En esa línea, cada palabra se puede expandir en una tupla de elementos que desglosen la información compactada en la forma superficial de las palabras. Por un lado, el lema o forma base, que viene a ser el término genérico que normalmente aparece en un diccionario académico, y, por otro, una etiqueta de características lingüísticas incluyendo información sobre su categoría sintáctica junto con una serie de propiedades como el género, el número, etc. De este modo, una definición tradicional de $\mathbf{s} = \mathbf{s}_1 \dots \mathbf{s}_J$ se reemplazaría mediante una cadena extendida $\mathbf{s}' = (\mathbf{s}_1, m_1, u_1) \dots (\mathbf{s}_J, m_J, u_J)$, donde m_j se referiría al lema de la palabra $\mathbf{s}_j$, cuya etiqueta de características lingüísticas fuere u_j.

Dado que un lema se puede ver como un *cluster* lingüístico, en el que se dan cita todas las palabras que comparten dicho lema, el vocabulario de traducción entre dos idiomas se puede ver reducido significativamente al sustituir las formas superficiales de las palabras por sus correspondientes formas base durante el aprendizaje de los modelos estadísticos de traducción.

Sean $\mathbf{s} = (s_1^J, m_1^J, u_1^J)$ y $\mathbf{t} = (t_1^I, n_1^I, v_1^I)$ una frase de entrada y de salida, respectivamente. Siguiendo un modelo generativo, existen diversas maneras de descomponer la distribución de probabilidad conjunta $\Pr(\mathbf{s}, \mathbf{t})$ mediante un esquema de categorización lingüística. Una de ellas puede ser la siguiente:

$$\begin{aligned}\Pr(\mathbf{s},\mathbf{t}) &= \Pr(s_1^J, m_1^J, u_1^J, t_1^I, n_1^I, v_1^I) \\ &= \Pr(m_1^J, n_1^I) \cdot \Pr(u_1^J | m_1^J, n_1^I) \cdot \\ &\quad \Pr(v_1^I | m_1^J, n_1^I, u_1^J) \cdot \\ &\quad \Pr(s_1^J | m_1^J, n_1^I, u_1^J, v_1^I) \cdot \\ &\quad \Pr(t_1^I | m_1^J, n_1^I, u_1^J, v_1^I, s_1^J)\end{aligned}$$

expresión que, bajo las siguientes asunciones:

1. $\Pr(m_1^J, n_1^I)$ se aproxima por la de su alineamiento óptimo $\Pr(m_1^J, n_1^I, \hat{a}_1^I)$
2. La etiqueta u es independiente de n, por lo que $\Pr(u_1^J | m_1^J, n_1^I) = \Pr(u_1^J | m_1^J)$
3. La etiqueta v no depende ni de m ni de n, tan sólo de su alineamiento $\hat{a}$, es decir, $\Pr(v_1^I | m_1^J, n_1^I, u_1^J) = \Pr(v_1^I | \hat{a}_1^I, u_1^J)$, y además dicha probabilidad es una función discreta binaria, de manera que:

$$\Pr(v_1^I | \hat{a}_1^I, u_1^J) = \begin{array}{ll} 1 & \text{si } \forall i = 1, \ldots, I : v_i = u_{\hat{a}_i} \\ 0 & \text{en caso contrario} \end{array}$$

   determinando las etiquetas de salida a partir de las de entrada.

4. Las palabras sólo dependen de sus lemas y etiquetas correspondientes:

$$\Pr(s_1^J | m_1^J, n_1^I, u_1^J, v_1^I) = \Pr(s_1^J | m_1^J, u_1^J)$$

$$\Pr(t_1^I | m_1^J, n_1^I, u_1^J, v_1^I, s_1^J) = \Pr(t_1^I | n_1^I, v_1^I)$$

finalmente queda como sigue:

$$\begin{aligned}\Pr(s_1^J, m_1^J, u_1^J, t_1^I, n_1^I, v_1^I) &\approx \Pr(m_1^J, n_1^I, \hat{a}_1^I) \cdot \Pr(u_1^J | m_1^J) \cdot \\ &\quad \Pr(v_1^I | \hat{a}_1^I, u_1^J) \cdot \Pr(s_1^J | m_1^J, u_1^J) \cdot \\ &\quad \Pr(t_1^I | n_1^I, v_1^I) \qquad (5.1)\end{aligned}$$

de modo que la expresión a resolver en traducción estadística se plantea así:

$$\begin{aligned}\hat{\mathbf{t}} &= \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{s}, \mathbf{t}) \\ &= \underset{t_1^I, n_1^I, v_1^I}{\operatorname{argmax}} \Pr(s_1^J, m_1^J, u_1^J, t_1^I, n_1^I, v_1^I)\end{aligned}$$

donde $\Pr(s_1^J, m_1^J, u_1^J, t_1^I, n_1^I, v_1^I)$ se aproxima por medio de la expresión de (5.1).

Dado que la maximización no depende de $\Pr(u_1^J|m_1^J)$, ni de $\Pr(s_1^J|m_1^J, u_1^J)$, y que, por la asunción 3, $\Pr(v_1^I|\hat{a}_1^I, u_1^J)$ presenta su valor máximo si $v_1^I = u_{\hat{a}_1^I}$, el problema de la búsqueda utilizando categorías lingüísticas se expresa así:

$$\hat{\mathbf{t}} \approx \underset{t_1^I, n_1^I}{\operatorname{argmax}} \Pr(m_1^J, n_1^I, \hat{a}_1^I) \cdot \Pr(t_1^I|n_1^I, u_{\hat{a}_1^I}) \tag{5.2}$$

La distribución de probabilidad conjunta basada en formas base se puede modelar adecuadamente mediante transductores GIATI, mientras que para la distribución de probabilidad $\Pr(t_1^I|n_1^I, u_{\hat{a}_1^I})$ se pueden usar una serie de diccionarios estocásticos especializados que revelen las palabras de salida desde sus correspondientes lemas, en función de ciertas etiquetas lingüísticas. Este comportamiento está basado en un sistema de traducción automática entre español y catalán [GLN+06] que asume que la información lingüística permanece inalterable entre la entrada y la salida en la mayoría de ocasiones.

Finalmente, la ecuación 5.2 se puede aproximar de manera subóptima así:

$$\begin{aligned}\hat{n}_1^I &= \underset{n_1^I}{\operatorname{argmax}} \Pr(m_1^J, n_1^I, \hat{a}_1^I) \\ \hat{t}_1^I &= \underset{t_1^I}{\operatorname{argmax}} \Pr(t_1^I|\hat{n}_1^I, u_{\hat{a}_1^I})\end{aligned} \tag{5.3}$$

lo que restringe la búsqueda para realizar primero una operación de transducción entre lemas, desde la entrada a la salida, para después convertir los lemas en palabras a través de las etiquetas de características correspondientes. En la Figura 5.1, se puede apreciar una arquitectura para esta aproximación, donde la entrada se preprocesa siguiendo sus propios criterios lingüísticos, definiendo individualmente a las palabras o unidades básicas de traducción.

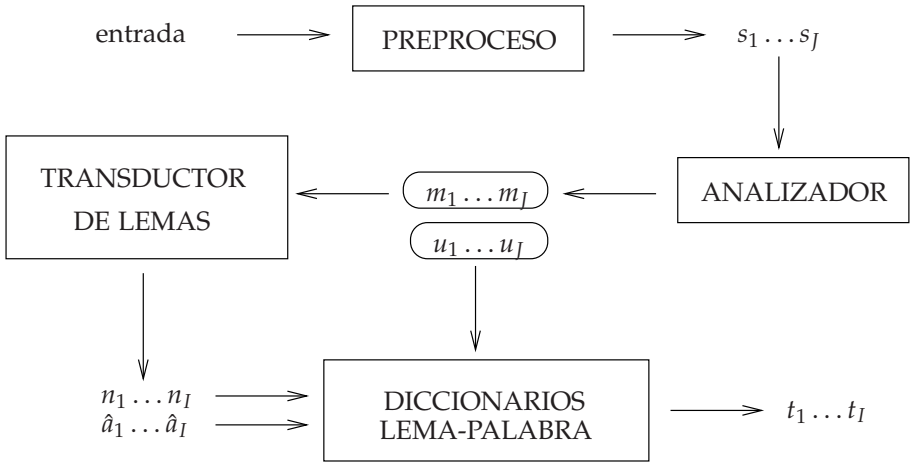


Figura 5.1: Arquitectura del sistema estadístico-morfológico

Para calcular la expresión $\Pr(t_1^I|\hat{n}_1^I, u_{\hat{a}_1^I})$, se emplean una serie de diccionarios estocásticos especializados cuya estimación se puede realizar siguiendo una aproximación basada en máxima verosimilitud. El criterio de especialización se puede ver desde dos puntos de vista equivalentes: por un lado, puede haber un diccionario estocástico para cada lema de salida, cuyos registros indicarían la probabilidad de generar una palabra de salida a partir de una etiqueta de características lingüísticas determinada; y, por otro lado, se puede construir un diccionario para cada etiqueta de características lingüísticas, que exprese la probabilidad de conversión entre lemas y palabras de salida. El cálculo de $\Pr(t_1^I|\hat{n}_1^I, u_{\hat{a}_1^I})$ se realiza por medio de la contribución de las probabilidades individuales entre cada *i*-ésimo lema y palabra de salida, junto con la etiqueta morfológica de la posición de entrada con la que se alinearon:

$$\Pr(t_1^I|\hat{n}_1^I, u_{\hat{a}_1^I}) \approx \prod_{i=1}^{I} \Pr(t_i|\hat{n}_i, u_{\hat{a}_i})$$

Los alineamientos estadísticos se suelen usar como parte del proceso de construcción de transductores, por lo tanto, durante la generación de los le-

mas de salida, es posible saber la posición de entrada con la que se alinearon.

5.3. Modelos probabilísticos

Como se ha expuesto en la sección 5.2, esta aproximación emplea dos tipos de modelos: por un lado, un transductor en el que los lemas de entrada se traducen a lemas de salida, junto con un vector de alineamientos que relaciona las posiciones de entrada de las que provienen; por otro lado, un conjunto de diccionarios estocásticos especializados que hacen emerger la forma superficial de las palabras de salida a partir de los correspondientes lemas y de la información morfológica de origen con la que están alineadas. Asimismo, un diccionario estocástico se puede representar por medio de un transductor estocástico de estados finitos, consiguiendo la homogeneización necesaria para permitir el empleo de operaciones entre los distintos modelos.

Este formalismo se introduce con la esperanza de mejorar el modelado de la transferencia de cadenas desde un idioma de entrada a otro de salida mediante las características de tipo morfológico comunes a todas las lenguas. Sin embargo, es evidente que partir el proceso de traducción en varias etapas implica un mayor número de oportunidades de que el sistema se equivoque. La sección 5.5 estudia los errores más frecuentes asociados a este heurístico.

5.3.1. Transductores GIATI basados en lemas

La implementación del esquema de categorización lingüístico propuesto requiere incrementar la información a incluir en cada símbolo extendido. Concretamente, cada lema de salida emitido por el modelo necesitará indicar la posición relativa de entrada de la que proviene a través de su alineamiento, es decir, la distancia entre la posición actual y aquella con la que está alineada.

La figura 5.2 muestra los dos tipos de situaciones en las que la función de etiquetado se puede ver envuelta a la hora de crear los símbolos extendidos.

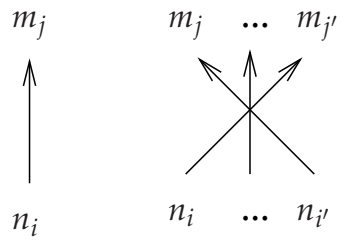


Figura 5.2: Dos tipos de alineamientos

Las relaciones como las del primer ejemplo (el alineamiento $n_i \rightarrow m_j$) establecen claramente transiciones del tipo m_j/n_i cuyo símbolo de salida n_i se produce sin retrasos respecto al símbolo de entrada m_j con el que se alinea. Esta sincronización se denota mediante un movimiento de retroceso nulo relativo a la posición actual de análisis, tal como se puede ver en la figura 5.3, con una representación directa de esta información sobre el modelo derivado.

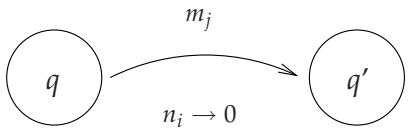


Figura 5.3: Símbolos extendidos uno-a-uno

Mientras que esta primera situación es indudablemente fácil de resolver, la segunda conlleva un poco más de trabajo. Un grupo de alineamientos cruzados provocaría generalmente retrasar la emisión de los símbolos de salida hasta que el análisis hubiera alcanzado progresivamente el último de los símbolos de entrada implicados. En este caso, no todos los lemas de salida se situarán en la misma arista que el símbolo de entrada con el que se alinean, tal como sucedía anteriormente, sino más bien a algunas aristas de distancia.

Como consecuencia, los lemas de salida se anotan conjuntamente mediante un indicador relativo de distancia respecto de la posición de entrada con la que se alinearon. Los elementos espúreos no necesitan de tal referencia debi-

do a su propia generación espontánea, independiente de cualquier elemento de entrada particular. En el segundo ejemplo de la figura 5.2, n_i se alinea con el símbolo actual de entrada $m_{j'}$, indicado como un desplazamiento nulo (0). Sin embargo, la emisión de $n_{i'}$ se retrasará, alejándose en el tiempo de su alineamiento de entrada m_j. Esta distancia relativa se anota entonces a continuación del símbolo de salida $n_{i'}$ como un enlace que permita posteriormente un movimiento de retroceso para acceder al símbolo lingüístico correspondiente. El resultado de tal algoritmo de etiquetado se puede apreciar directamente sobre el transductor final que se obtendría, tal como la figura 5.4 lo muestra.

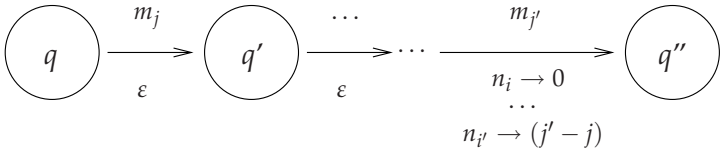


Figura 5.4: Símbolos extendidos uno-a-muchos

Nótese que la distancia relativa para el símbolo $n_{i'}$ se calcula como la diferencia entre la posición de análisis j' y la posición con la que se alinea, j.

La incorporación de los alineamientos internos en los símbolos extendidos genera una especialización de éstos en función de su patrón de alineamiento. Lo que, de por sí, no es un fenómeno excesivamente frecuente en traducción, de modo que la inclusión de esta información no añade mayor incertidumbre. No obstante, esta aproximación está restringida mediante el uso de una función de alineamiento que debe ser, en todo caso, una relación uno-a-muchos, lo que impide su interacción con alineamientos no restringidos a una función que pudieran modelar relaciones plurales y recíprocas en las dos direcciones.

5.3.2. Diccionarios de conversión entre lemas y palabras

Un diccionario *ponderado* es una colección $(a, b, W(a, b))$ que contiene un conjunto de símbolos de traducción junto con un indicador de su fiabilidad.

Si $W(a,b) = \Pr(a|b) \wedge \forall y \sum_x \Pr(x|y) = 1$, entonces se le denomina *estocástico*.

Una vez que una frase de entrada lematizada se ha procesado a través de un modelo de transducción, la salida se expresa asimismo como una secuencia de lemas. Éstos se pueden transformar en palabras por medio de una serie de diccionarios estocásticos especializados que tienen en cuenta la información lingüística de los elementos de la entrada con los que están conectados.

Siguiendo el criterio de máxima verosimilitud, se puede estimar un diccionario de estas características mediante el conteo de frecuencias normalizadas:

$$\Pr(t_i|n_i) = \frac{F(t_i, n_i)}{\sum_x F(x, n_i)}$$

cuya implementación está integrada en el software desarrollado en esta tesis.

Estos diccionarios se pueden aprender por medio de dos métodos de estimación diferentes: por un lado, se puede considerar únicamente un corpus monolingüe de salida, por lo que el aprendizaje de la morfología se realiza a través de la misma lengua; y, por otro lado, los alineamientos estadísticos entre un corpus bilingüe pueder permitir el entrenamiento de estos diccionarios usando la información lingüística de entrada como factor de especialización. Un esquema de este último método se describe gráficamente en la figura 5.5.

En este caso, los alineamientos que son necesarios para el aprendizaje de los transductores GIATI basados en lemas son igualmente apropiados para la construcción de diccionarios probabilísticos entre lemas y palabras de salida.

5.3.3. Arquitectura integrada mediante composición al vuelo

Las ecuaciones (5.3) representan la estrategia de búsqueda para traducir una frase en lenguaje natural desde un idioma de entrada hacia uno de salida. Según estas ecuaciones, el proceso de traducción se realiza en dos etapas: primeramente, la secuencia de lemas de entrada se traduce a lemas de salida por medio de una aproximación basada en transductores de estados finitos;

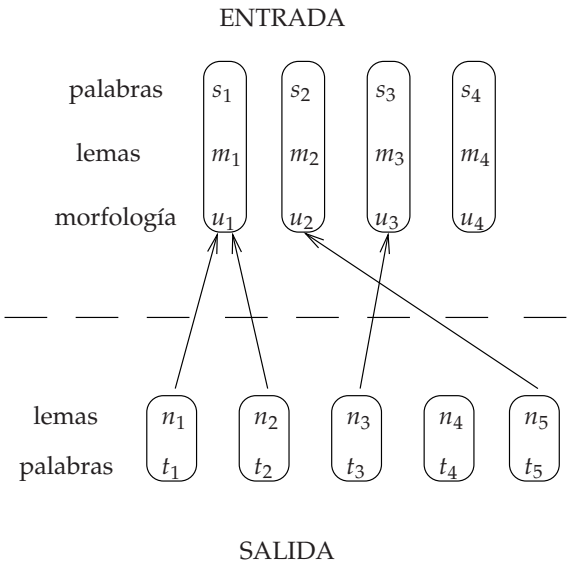


Figura 5.5: Usando las etiquetas u para una estimación bilingüe de $\Pr(t_i|n_i)$

seguidamente, los lemas de salida se convierten en palabras de salida a través de un conjunto de diccionarios estocásticos especializados lingüísticamente.

Sin embargo, este procedimiento en dos fases se puede integrar en un único proceso que no resulte subóptimo, abordando así la ecuación (5.2) de forma directa, componiendo la salida del primer transductor con la entrada de los diccionarios, cuya representación por medio de transductores es posible. La composición es una operación de clausura sobre transductores de estados finitos, aplicable sobre sus extensiones por medio de pesos o probabilidades. Sean Σ, Φ, y Δ, tres alfabetos definidos mediante el conjunto de sus símbolos. Si $\mathcal{T}_1$ es un transductor que modela relaciones en el conjunto $\Sigma^\star \times \Phi^\star$, y $\mathcal{T}_2$ es un transductor que lo hace sobre $\Phi^\star \times \Delta^\star$, entonces la composición entre ambos, $\mathcal{T}_1 \circ \mathcal{T}_2$, es un transductor cuya relación se incluye en $\Sigma^\star \times \Delta^\star$ [JM08]. Algebraicamente, aplicar una secuencia de entrada m_1^J sobre $\mathcal{T}_1 \circ \mathcal{T}_2$ produce el mismo resultado que utilizar los dos transductores originales en cascada, es decir, aplicar m_1^J sobre $\mathcal{T}_1$, y el resultado de esta operación, n_1^I, sobre $\mathcal{T}_2$.

Dado que la composición de transductores suele producir un modelo más complejo, además de que dicha composición no se puede realizar a priori debido a que depende en tiempo real del análisis morfológico de entrada u_1^J, la solución comúnmente adoptada consiste en hacer la composición al vuelo, aplicando el diccionario correspondiente $\mathcal{T}_2$ sobre cada uno de los lemas de salida que aparecen como parte de una hipótesis parcial durante el análisis de m_1^J sobre el transductor de lemas $\mathcal{T}_1$, mediante el uso de sus transiciones.

De este modo, se puede abordar directamente el problema planteado en la ecuación 5.2 sin tener que maximizar dos procesos distintos por separado. Dado que la información de alineamiento se refiere siempre a un símbolo de entrada analizado previamente, su etiqueta lingüística asociada estará siempre disponible para poder utilizar el diccionario correspondiente $\Pr(t_i|n_i, u_{\hat{a}_i})$.

Gracias a la inclusión de la información de alineamiento entre los símbolos de salida, es posible saber para cada lema generado, la posición de entrada con la que se conecta. Como resultado, cada lema producido como parte de una hipótesis parcial de salida se puede convertir y almacenar como una palabra de salida sin la necesidad de tener que esperar a que la mejor hipótesis de salida $\hat{n}_1^I$ se haya generado completamente. Una vez se ha analizado la secuencia de entrada m_1^J a través del transductor GIATI de formas base, la frase de salida $\hat{\mathbf{t}}_1^I$, ya en su forma superficial, se encuentra disponible.

5.4. Experimentos

Se ha realizado un conjunto de experimentos de carácter preliminar para evaluar la viabilidad de nuestra aproximación integrada de traducción automática basada en categorías lingüísticas, mediante un análisis morfológico.

Para el diseño de la configuración experimental, se han utilizado dos tareas de grados de dificultad muy diferentes. La tarea EuTrans [ABC+00] se define en el dominio restringido en el contexto de un diálogo en el mostrador de un hotel. Es una tarea generada artificialmente a partir de una serie de esquemas dirigidos por la sintáxis. Las características del corpus EuTrans

se pueden observar en la tabla 5.1. Los experimentos se han realizado sobre la dirección de traducción español→inglés en esta tarea de baja perplejidad. EuTrans es una tarea de traducción que a menudo se utiliza con propósito de depuración. Generalmente, las nuevas aproximaciones a la traducción automática estadística se evalúan en primer lugar mediante esta tarea sin complejidad aparente para establecer algún tipo de criterio de comportamiento.

Tabla 5.1: Características del corpus EuTrans

		español	**inglés**
Entrenamiento	Nº frases	10000	
	Nº palabras	97.1 K	99.3 K
	Vocabulario	686	513
Test	Nº frases	3000	
	Nº palabras	35.1 K	35.6 K
	Perplejidad	4.9	3.6

Por otro lado, esta aproximación se ha aplicado también sobre una sección portugués–español del corpus del EuroParl. Sus características se pueden analizar en la tabla 5.2.

Tabla 5.2: Características del corpus portugués-español del EuroParl

		portugués	**español**
Entrenamiento	Nº frases	50000	
	Nº palabras	1.3 M	1.3 M
	Vocabulario	37.3 K	37.6 K
Test	Nº frases	1000	
	Nº palabras	25.4 K	26.0 K
	Perplejidad	121.3	103.5

La lematización del corpus EuTrans y su etiquetado morfológico se ha realizado a través de la herramienta FreeLing [ACC+06], mientras que, para

el análisis de las frases en español del EuroParl, se utilizó SisHiTra [GLN+06]. Los lemas portugueses y sus etiquetas de características lingüísticas fueron proporcionados por el Laboratorio de Sistemas de Lengua Hablada del Instituto de Ingeniería de Sistemas y Computadores I+D de Lisboa. Ambos corpus se alinearon a nivel de palabra mediante el sofware estadístico GIZA++.

Diversas opciones de preproceso (también conocido como tokenización) fueron evaluadas para establecer un punto de partida en el que aplicar el esquema de categorización lingüístico propuesto en esta tesis.

5.4.1. Resultados de traducción

El análisis morfológico realizado sobre EuTrans se resume en la tabla 5.3. La variación sobre vocabularios y perplejidades es prácticamente inexistente por lo que no es de extrañar que este método no tenga utilidad en esta tarea.

Tabla 5.3: EuTrans: vocabularios de entrenamiento y perplejidades del test. Se comparan tres corpus diferentes: a) el derivado de la partición original; b) aplicando un preproceso basado en técnicas lingüísticas; c) utilizando el marco de categorización propuesto a través de la morfología de las palabras.

	Vocabulario		**Perplejidad**	
Corpus	español	inglés	español	inglés
Baseline	686	513	4.9	3.6
Preproceso	624	513	5.2	3.6
Categorización	476	503	4.6	3.6

Las tasas obtenidas confirman la degradación del rendimiento del sistema, aunque las técnicas de preproceso por sí solas muestren una cierta mejora. La sencillez de este corpus es muy probablemente la razón de tal resultado, lo que se verifica a través de los indicadores que se reportan en la tabla 5.3, junto a las diferencias que sufren ambas lenguas con la utilización de clases.

De hecho, si evaluamos el resultado del transductor basado en categorías,

es decir, obviando del sistema la conversión entre lemas y palabras de salida, comparando de esta manera hipótesis y referencias compuestas de categorías, las tasas tampoco mejoran las de los sistemas basados sólamente en palabras. Este paradójico resultado se explica mediante un ejemplo extraído del corpus: en un sistema clásico, las palabras *quiero* y *querría* se identifican y se traducen correctamente por las expresiones inglesas *I want* y *I'd like*, respectivamente. Sin embargo, al utilizar lemas, las palabras de entrada se clasifican dentro de la misma clase como formas del verbo *querer*, pero no sucede así con las de salida, cuya clasificación se produce sobre categorías diferentes (*want* y *like*), introduciendo una ambigüedad de traducción que anteriormente no existía.

Por otro lado, EuroParl es una tarea más compleja, lo que se refleja a través de sus indicadores de vocabulario y perplejidad. La lematización reduce los vocabularios prácticamente a la mitad, provocando a su vez una caída significativa en las perplejidades, tal y como se puede apreciar en la tabla 5.4. Los resultados sobre el corpus del parlamento europeo están en la tabla 5.5.

En este caso, utilizar un corpus anotado morfológicamente ayuda al proceso de traducción. Además del preproceso, el método de categorización también permite un mejor modelado de las relaciones de transferencia entre los idiomas de entrada y salida. El tamaño de los modelos se ve asimimo significativamente reducido, lo que redunda no sólo en una estimación más robusta de parámetros durante la etapa de entrenamiento sino también en un ahorro en el consumo de recursos espacio-temporales durante la etapa de búsqueda.

Para encontrar la mejor configuración del sistema basado en categorías, hay que explorar el rendimiento de los dos módulos que se ven implicados. Por un lado, el transductor que transfiere lemas de entrada a lemas de salida es un modelo cuyas características se han descrito en los Capítulos 2 y 3 y cuyos criterios de optimización se detallan igualmente entre sus contenidos.

Por otro lado, los diccionarios que convierten lemas de salida en palabras se pueden estimar mediante dos métodos introducidos en la sección 5.3.2, denominados como entrenamiento monolingüe y bilingüe, respectivamente, que difieren en el origen de las etiquetas usadas para identicar a los modelos.

Tabla 5.4: EuroParl: vocabularios de entrenamiento y perplejidades del test. Se comparan tres sistemas diferentes: a) el que se deriva del corpus original; b) aplicando un preproceso basado en técnicas lingüísticas; c) utilizando el marco de categorización propuesto a través de la morfología de las palabras.

	Vocabulario		**Perplejidad**	
Sistema	portugués	español	portugués	español
Baseline	37.3 K	37.6 K	121.3	103.5
Preproceso	37.3 K	37.5 K	121.3	120.9
Categorización	18.3 K	19.3 K	91.1	91.1

Tabla 5.5: Resumen de resultados de traducción sobre el corpus del EuroParl. Se muestran las tasas obtenidas por medio de los tres sistemas de la tabla 5.4 junto con el número de estados y de aristas de los modelos de traducción, desglosando este último, si aplicable, entre los distintos modelos empleados.

	Tasas		**Tamaño de los modelos**	
Sistema	BLEU	WER	Nº estados	Nº aristas
Baseline	19.8	67.8	205 K	1.06 M
Preproceso	**20.0**	**65.7**	200 K	1.04 M
Categorización	**21.4**	**63.2**	166 K	925 K + 94 K

Lamentablemente, dada la combinación de utilidades lingüísticas usada, cuyas etiquetas morfológicas no son mayoritariamente compatibles entre sí, el uso de modelos entrenados mediante estimación monolingüe no ha lugar. Esto resulta comprensible en el caso del EuroParl ya que se emplearon dos herramientas lingüísticas completamente independientes para el etiquetado. Sin embargo, para la tarea EuTrans, se utilizó el mismo software para los dos idiomas, por lo que resulta un tanto decepcionante que no exista una unificación de etiquetas morfológicas para todas las lenguas que puede analizar.

Concluyendo, la aplicación de esta aproximación híbrida, combinando tecnología estadística y lingüística, sobre tareas de una cierta envergadura, permite un mejor modelado en traducción al de un sistema estadístico puro. No obstante, hay que hacer constar que esta tecnología es un tanto preliminar, cuyo éxito contrastado con otras aproximaciones aún está lejos de producirse. De hecho, las cifras oficiales con el corpus portugués-español del EuroParl, aunque hayan usado para ello un conjunto mucho mayor de entrenamiento, son decididamente mejores. Los experimentos mediante esta aproximación, lejos de pretender competir con otros sistemas, permiten establecer las bases y el escenario de una línea post-doc de investigación, ilustrando su potencial.

5.5. Análisis de errores

Para comprender mejor los puntos fuertes y débiles de esta aproximación, hemos analizado los resultados en cada etapa del prototipo de traducción sobre algún ejemplo del sentido portugués→español del corpus del EuroParl.

En la tabla 5.6, la serie de pasos que producen un resultado de traducción. Como se puede apreciar, el error más importante independiente de referencia es la omisión de **deberíamos preocuparnos** en la hipótesis de traducción t, que realmente es debida al transductor GIATI, ya que n tampoco la refleja. Representa un error directamente atribuible al empleo de esta metodología ya que mediante una aproximación directa a la traducción éste no se produce. Por otro lado, vemos como la concordancia de sustantivos y verbos es correcta

basándose en la morfología de las palabras de origen con las que se alinean. Discusión aparte es el hecho de que **as orientações** en la frase de origen aparece traducido en la referencia de traducción en singular: **la orientación**. Dicho fenómeno es difícil de capturar por medio de esta clase de tecnología excepto si esta discrepancia se produce con alta frecencia en dichas palabras. Otro tema es si mantener el plural puede considerarse una buena traducción, lo que desde un punto de vista pragmático, todo tiende a indicar que así es.

Tabla 5.6: Análisis de los procesos intermedios que suceden en traducción. La frase original en portugués es la marcada como **s**, su lematización es *m*, la traducción de *m* a través de un transductor GIATI basado en lemas es *n*, finalmente, la recuperación de la forma superficial de salida se observa en *t*. También se incluye el resultado directo **s** → **t** y la referencia de traducción **r**.

s	todavia , entendo que deveríamos preocupar-nos ainda mais com as orientações e com os resultados da política regional da comunidade .
m	todavia , entender que dever preocupar ainda mais com as orientação e com os resultado da política regional da comunidade .
n	sin+embargo , @considerar que aún más con las orientaci-f13 y con los resultado-f4 de la polític-f28 regional-f20 de la comunidad-f5 .
t	sin embargo , considero que aún más con las orientaciones y con los resultados de la política regional de la comunidad .
t	sin embargo , considero que deberíamos preocupar-nos aún más con las orientaciones y con los resultados de la política regional de la comunidad .
r	opino , no obstante , que deberíamos mostrar más preocupación por la orientación y los resultados de la política regional de la comunidad .

En la tabla 5.7, otro ejemplo extraído del uso de morfología en traducción. En esta ocasión, se observa un error en la forma verbal **es** derivada de **@ser**, motivado muy probablemente por un alineamiento erróneo, condicionando de esta manera una etiqueta lingüística diferente con la que debe concordar. La palabra correcta la encontramos en la referencia de traducción dada: **sido**. Sin embargo, en líneas generales se aprecia una traducción de mayor calidad que la que se obtiene para este ejemplo a través de una aproximación directa.

Tabla 5.7: Análisis de los procesos intermedios que suceden en traducción. Una formación errónea de palabras a partir de sus lemas se rotula en negrita.

s	por outras palavras , este procedimento não teria **sido** necessário , se a comissão tivesse actuado na altura devida .
m	por outras palavra , este procedimento não ter **ser** necessário , se a comissão ter actuar na altura dever .
n	@decir de+otro+modo , **@ser**+necesario , si la comisi-f13 @haber @actuar @llegar el momento-f4 .
t	dicho de otro modo , **es** necesario , si la comisión hubiera actuado llegado el momento .
t	en otras palabras , si la comisión tuviese hubiera trabajado llegado el momento .
r	en otras palabras , este procedimiento no habría **sido** necesario si la comisión hubiera actuado a tiempo en su momento .

En la tabla 5.8, una muestra más del empleo de morfología en traducción. Nuevamente, observamos algunos errores bien en la transferencia de lemas, bien en la generación de las formas superficiales de palabras a partir de éstos. En cualquier caso, de nuevo la arquitectura basada en información lingüística proporciona una solución mejor que la de un proceso directo de traducción.

En general, la impresión sobre esta arquitectura de traducción es positiva. Sin embargo, no cabe duda que esta aproximación es todavía muy prematura y que representa solamente el inicio de una futura línea de investigación,

Tabla 5.8: Análisis de los procesos intermedios que suceden en traducción.

s	a defesa do consumidor exige que haja uma informação muito precisa e completa sobre os alimentos , incluindo a relativa aos novos alimentos para animais e à rotulagem de produtos isentos de ogm .
m	a defesa do consumidor exigir que haver uma informação muito precisar e completo sobre os alimento , incluir a relativa aos novo alimento para animal e à rotulagem de produto isento de ogm .
n	la protecci-f13 de el consumidor-f21 @exigir que hay un-f48 informaci-f13 precis-f28 y complet-f28 sobre los alimento-f4 , @incluir la @destinar a los nuev-f28 animal-f20 y a el etiquetado-f4 de producto-f4 libre-f23 de omg .
t	la protección del consumidor exige que hay una información precisa y completa sobre los alimentos , incluidos la destinada a los nuevos animal y al etiquetado de productos libre de omg .
t	pide el consumidor muy precisa y completa sobre los alimentos - y también la sobre los nuevos y al etiquetado de productos .
r	la defensa del consumidor exige que haya una información muy precisa y completa sobre los alimentos , incluida la relativa a los nuevos alimentos para animales y al etiquetado de productos exentos de ogm .

cuya motivación esperamos haber conseguido transmitir desde estas páginas.

5.6. Resumen del capítulo

Este capítulo ha introducido una aproximación a la traducción estadística que explota la morfología de las palabras bajo un entorno de estados finitos.

En ese sentido, el marco propuesto se basa en el uso de clases lingüísticas de manera que las palabras se agrupan en función de su origen morfológico.

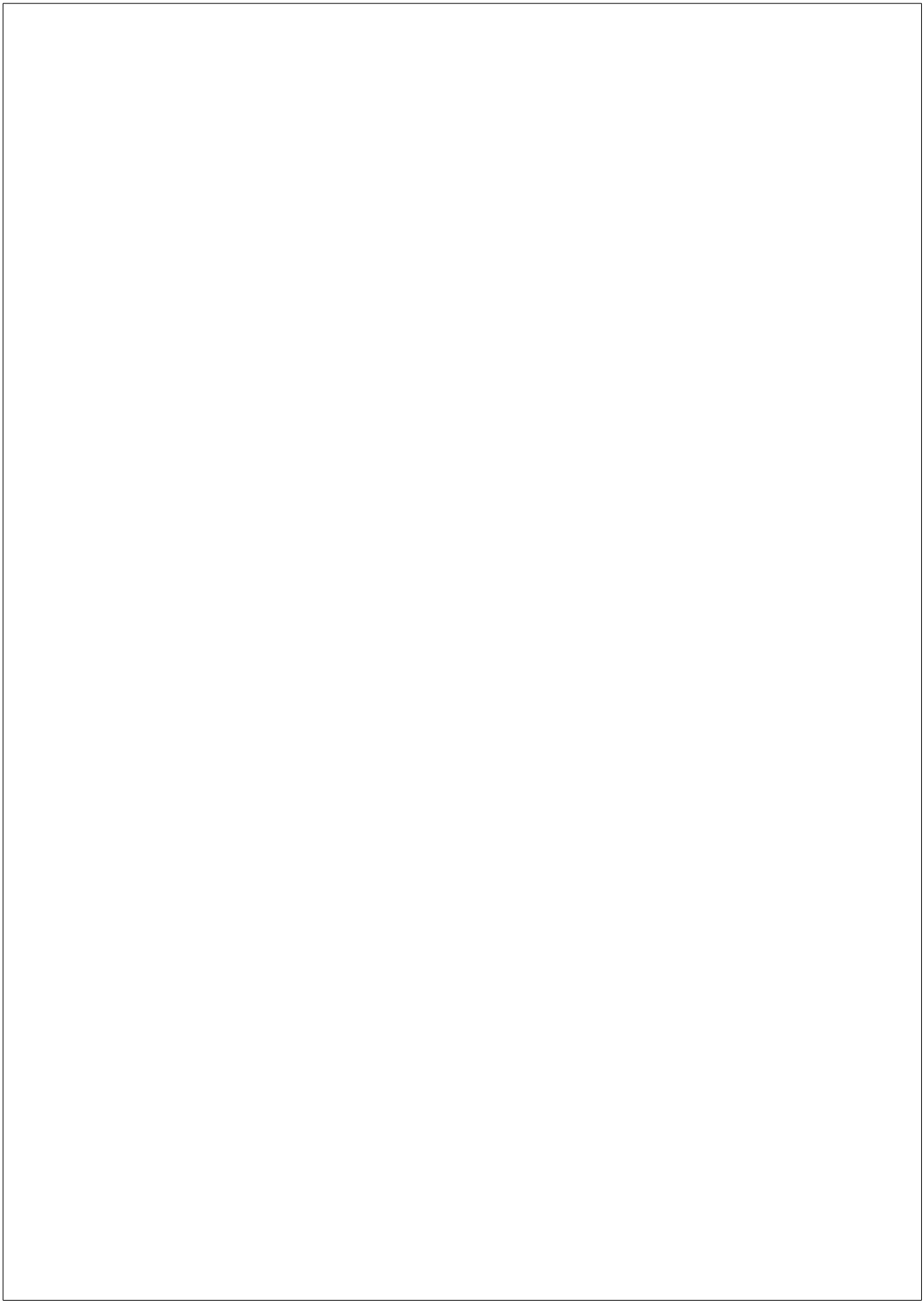
Para ello, se ha introducido una metodología de traducción de dos modelos donde la transferencia de significado se realiza a nivel de formas base o lemas para luego generar las palabras a partir de éstos mediante métodos de flexión.

Dicha infraestructura se implementa mediante transductores estocásticos, a través de la integración de la información de alineamiento en los modelos, permitiendo el acceso a las etiquetas lingüísticas con las que deben concordar. La búsqueda de la traducción se realiza mediante una composición *on-the-fly* de los modelos requeridos según indique el análisis morfológico de entrada.

La experimentación realizada permite afrontar halagüeñamente el futuro, habiendo obtenido cierta perspectiva inicial sobre este tipo de aproximación por medio de un pequeño análisis de los errores más frecuentes acontecidos.

Agradecimientos

Nos gustaría agradecer a las personas del Laboratorio de Sistemas de Lengua Hablada del Instituto de Ingeniería de Sistemas y Computadores I+D de Lisboa, y de forma especial a Diamantino Caseiro, por haber proporcionado el análisis morfológico de la partición portuguesa del corpus del EuroParl.



Capítulo 6

Conclusiones

La principal contribución de esta tesis es a través de los transductores basados en segmentos, cuya definición proviene de un modelo de alineamiento muchos-a-muchos que representa el estado del arte en traducción estadística por medio de los así llamados modelos de traducción basados en segmentos.

La mayoría de técnicas que emplean este tipo de modelos, no obstante, consumen una cantidad excesiva de recursos computacionales en ejecución, reduciendo su efectividad como motores de traducción *online* en tiempo real. Un marco de trabajo basado en modelos de estados finitos representa una alternativa interesante en la que desarrollar modelos basados en segmentos ya que constituye un paradigma eficaz en el que factores de calidad y de eficiencia se integran adecuadamente en el diseño y construcción de dispositivos de traducción que puedan llegar a ser útiles a sus potenciales usuarios.

La idea de usar diccionarios basados en segmentos (frente a los basados en palabras) puede también trasladarse a los transductores de estados finitos. En esta tesis se han descrito los detalles de implementación que son necesarios para la construcción de transductores GIATI basados en segmentos a partir de una segmentación bilingüe y monótona (o de una aproximación) del corpus de entrenamiento. Además, también se han descrito en profundidad los fenómenos algorítmicos que se deben tener en cuenta durante la fase de

descodificación a través de este tipo de transductores basados en segmentos.

En este marco, hemos desarrollado una serie de herramientas software que implementan de manera eficiente tanto la construcción de transductores estocásticos mediante la metodología GIATI como el proceso de búsqueda a través de este tipo de modelos, utilizando *C* como lenguaje de programación. Todo ello ha dado lugar al nacimiento de un *toolkit* de estados finitos de nombre GREAT (*GIATI and Refx Enhanced via Annotation Techniques*) y que utiliza técnicas de escalabilidad para su uso con grandes volúmenes de datos.

La eficiencia computacional, tanto espacial como temporal, es uno de los problemas que algunas implementaciones previas presentan de manera evidente, especialmente si el volumen de datos es de una cierta relevancia. Las mejoras no se deben exclusivamente al haber abandonado *Python* como lenguaje de programación sino también al refinamiento en la lógica de los algoritmos, así como a las técnicas de escalabilidad propuestas en forma de filtrado de transductores, que mantienen el modelo estimado como un conjunto de n-gramas, expresando mediante una red de estados finitos sólo aquellos que hacen falta para traducir un determinado corpus de evaluación.

GREAT emplea asimismo lo que se conoce como modelos factoriales, reduciendo la variabilidad de los textos en lenguaje natural mediante la incorporación de información lingüística en los modelos de naturaleza estadística.

Los modelos log-lineal representan en la actualidad una opción bien establecida en traducción automática cuya naturaleza reside en la integración de una serie de modelos heterogéneos bajo una única aproximación estadística. GREAT es capaz de modelar una combinación log-lineal de diversos modelos locales por medio de su expresión como transductores basados en segmentos.

GREAT implementa a su vez un nuevo algoritmo de suavizado basado en los mismos modelos de n-gramas con *backoff*, que se reduce a una búsqueda restringida mediante la gestión dinámica de una lista de estados prohibidos.

Asimismo, GREAT implementa estrategias de búsqueda que se adaptan a la granularidad de los modelos, permitiendo realizar análisis basados en

segmentos en aquellos transductores de este tipo, aumentando la eficacia y la eficiencia del algoritmo de descodificación de Viterbi con búsqueda en haz.

Con respecto a los resultados de experimentación, se demuestra que la propuesta de modelización a través de transductores basados en segmentos es muy efectiva. En ellos se observa con claridad que estos transductores superan ampliamente a los modelos pioneros de GIATI, basados en palabras.

Además, el diseño de transductores basados en otros modelos de traducción se ha revelado como una aproximación eficiente a los sistemas de traducción automática estado del arte, bajo la que implementar adecuadamente entornos de traducción con demanda de rendimiento en tiempo real.

Por último, las técnicas propuestas para la aceleración de la búsqueda a través de transductores GIATI han demostrado ser efectivas, tanto mediante la poda de modelos que reduce su tamaño como a través del uso de estrategias de descodificación basadas en la propia naturaleza de los modelos.

En cuanto al resto del trabajo realizado, aunque de menor repercusión, hemos logrado formalizar un marco en el que integrar una modelización log-lineal mediante la combinación de diversos transductores de estados finitos. De aplicación sobre los denominados transductores basados en segmentos, cada modelo local otorga un valor de función sobre cada par de segmentos. La optimización de pesos se realiza mediante un algoritmo de descenso por gradiente sobre un transductor extendido que contempla la información de todos los modelos a la vez. Una vez encontrados los valores óptimos, una sencilla operación de *rescoring*, aplicada sobre el transductor GIATI original, permite condensar en un solo modelo la información de todos ellos (y de sus respectivos pesos), con un tamaño que incluso puede haberse visto reducido.

Finalmente, hemos incluido una aproximación a la traducción automática a través del uso de categorías lingüísticas en transductores de estados finitos. El modelo de traducción entre lemas y el conjunto de diccionarios que transfieren el idioma destino a lenguaje natural se combinan en una arquitectura integrada mediante composición al vuelo de transductores de estados finitos. Los resultados son preliminares y con unas limitaciones muy evidentes, sin

embargo, la idea de una aproximación híbrida en traducción automática, donde estadística y lingüística puedan cooperar para lograr el ansiado éxito, resulta muy atractiva desde un punto de vista científico, lo que la convierte en una línea de investigación abierta de cara al futuro.

6.1. Futuras líneas de trabajo

En cuanto al trabajo futuro asociado a esta tesis, existen diversas líneas de investigación con las que continuar el estudio presentado en este documento. Por un lado, está la ya mencionada sobre la incorporación de un componente morfológico en el proceso de traducción que resulte eficaz a todas luces. Esta vía no está exclusivamente asociada a una metodología de estados finitos, sino que representa un reto intelectual aún por resolver en traducción estadística cuyo éxito supondría un impacto de inmensas proporciones.

La incorporación del resto de modelos habituales del escenario log-lineal, junto con el estudio de una posible implementación de modelos jerárquicos, son otras aplicaciones de futuro a desarrollar bajo el entorno de GREAT, cuya funcionalidad se ampliará también para ofrecer los medios de obtener resultados de tipo *n-best* en forma de cadenas o mediante grafos de palabras.

Por otro lado, una de las conclusiones más importantes de esta tesis se refiere a la variabilidad en las prestaciones del sistema en función de la segmentación bilingüe y monótona que extraigamos del conjunto de entrenamiento. Para ser competitivos y disputar el estado del arte, al que nos hemos acercado mucho en el desarrollo de esta tesis, una vía a explorar es la de investigar técnicas de segmentación en un contexto de traducción que puedan mejorar la situación actual, basada en el uso heurístico de alineamientos estadísticos.

Sistemas como Moses, cuyo modelo de traducción ha podido utilizarse simplificadamente en el contexto de GIATI, optan sin embargo por una extracción masiva de pares de segmentos a partir de susodichos alineamientos, cuyo modelado conduce no a una única segmentación bilingüe por cada par de frases paralelas sino a un conjunto virtual de numerosas segmentaciones.

Esta es una línea que habría igualmente que investigar, y que nos llevaría directamente a plantearnos nuevas técnicas de escalabilidad que permitiesen compactar una información que por sí sola es altamente redundante.

Por último, dado que la unidad básica en traducción es el segmento, modelando internamente el contexto en el que aparecen todas sus palabras, quizá una dirección más de investigación que asimismo se debiera explorar sea el desarrollo de un esquema de reordenamiento de segmentos bajo GIATI, lo que [KDB06] ha incorporado con éxito en su arquitectura de transductores.

6.2. Publicaciones de investigación

Las publicaciones producidas en un contexto directamente relacionado con esta tesis se deben a diversas contribuciones a congresos internacionales, cuya correspondencia con los contenidos de la misma se cita entre paréntesis:

- J. González and F. Casacuberta. GREAT: a finite-state machine translation toolkit implementing a Grammatical Inference Approach for Transducer Inference. In *Proceedings of the EACL 2009 Workshop on Computational Linguistics Aspects of Grammatical Inference*, pages 24–32, Athens, Greece, 30 March 2009. (Secciones 3.2.1, 3.4, 4.3.2 y 4.3.4).

- J. González and F. Casacuberta. A finite-state framework for log-linear models in Machine Translation. In *Proc. of the 12th European Association for Machine Translation Conference*, pages 41–46, Hamburg, Germany, 22-23 September 2008. (Secciones 3.5 y 4.3.5).

- J. González, G. Sanchis, and F. Casacuberta. Learning finite state transducers using bilingual phrases. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics. Lecture Notes in Computer Science*, volume 4919, pages 411–422, Haifa, Israel, February 17 to 23 2008. (Secciones 3.3.2 y 4.3.3).

- J. González and F. Casacuberta. Linguistic Categorisation in Machine Translation using Stochastic Finite State Transducers. In *Mixing Approaches to Machine Translation*, San Sebastian, Spain, 14/02/2008. (Cap. 5).
- J. González and F. Casacuberta. Phrase-based finite state models. In *Proceedings of the 6th International Workshop on Finite State Methods and Natural Language Processing (FSMNLP)*, Potsdam, Germany, September 14-16 2007. (Secciones 3.3 y 4.3.1).

Otros trabajos publicados por el autor como telón de fondo de esta tesis, relacionados con traducción automática mediante modelos de estados finitos:

- D. Picó, J. González, F. Casacuberta, D. Caseiro, and I. Trancoso. Finite-state transducer inference for a speech-input Portuguese-to-English machine translation system. In *Proceedings of Interspeech'05*, Lisboa, Portugal, September 2005.
- J. Civera, J. M. Vilar, E. Cubel, A. L. 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.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
- 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 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona, Spain, 2004.
- E. Cubel, J. Civera, J. M. Vilar, A. L. Lagarda, 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 (ECAI04)*, pages 586–590, Valencia, Spain, August 2004. IOS Press.

- J. González, D. Ortiz, J. Tomás, and F. Casacuberta. A comparison of different statistical machine translation approaches for Spanish-to-Basque translation. In *Actas de las Terceras Jornadas en Tecnologías del Habla*, Valencia, Spain, November 2004.

- A. Pérez, J. González, F. Casacuberta, and I. Torres. Traducción automática mediante transductores estocásticos de estados finitos. In *Actas de las Terceras Jornadas en Tecnologías del Habla*, Valencia, Spain, November 2004.

- 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*, Dublin, Ireland, 2003.

- D. Ortiz, I. García-Varea, F. Casacuberta, J. González, and A. Lagarda. On the use of statistical machine-translation techniques within a memory-based translation system (AMETRA). In *Ninth Machine Translation Summit*, pages 299–306. AMTA, New Orleans, US, September 2003.

Otros trabajos publicados por el autor como telón de fondo de esta tesis, relacionados con traducción automática utilizando conocimiento lingüístico:

- Jorge González, A. Giménez, Jesús González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, V. Félix, P. Peris, and F. Casacuberta. Una evaluación exhaustiva de SisHiTra, un paradigma híbrido en Traducción Automática. In *IV Jornadas en Tecnologías del Habla (IVJTH'2006)*, Zaragoza, Spain, Nov 2006.

- J. González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, A. Giménez, F. Casacuberta, J. M de Val, and F. Fabregat. SisHiTra: a Spanish-to-Catalan hybrid machine translation system. In *5th SALTMIL Workshop on Minority Languages*, pages 69–73, Genoa, Italy, 23 May 2006.

- J. R. Navarro, J. González, D. Picó, F. Casacuberta, J. M. de Val, F. Fabregat, F. Pla, and J. Tomás. SisHiTra: A hybrid machine translation system from Spanish to Catalan. In J. L. Vicedo, P. Martínez-Barco, R. Muñoz, and M. Saiz, editors, *Advances in Natural Language Processing*, Lecture Notes in Artificial Intelligence, pages 349–359. Springer-Verlag, Alicante, Spain, 2004.

- J. González, F. J. Nevado, J. R. Navarro, M. Pastor, F. Casacuberta, E. Vidal, F. Fabregat, J. M. de Val, L. Arenas, F. Pla, and J. Tomás. SisHiTra: Sistemas de traducción catalán-castellano y castellano-catalán con entrada de texto y voz. In *II Jornadas en Tecnologías del Habla*, Granada, Spain, December 2002.

Apéndice A

Resultados experimentales en detalle

Las tablas de este apéndice muestran el resultado de una experimentación exhaustiva desarrollada en el marco de esta tesis, en la que se ha explorado con ahinco el efecto de una serie de parámetros descritos en este documento.

Se explora el efecto de los así llamados transductores basados en segmentos, cuyo origen se analiza en ocasiones bajo diversas configuraciones, bien mediante otros modelos de traducción o usando alineamientos estadísticos. Los resultados se comparan con los de los transductores basados en palabras, considerando estos últimos como sistemas base (*baseline*) a todos los efectos.

Asimismo, se explora la aportación en materia de suavizado que consiste en la interpretación del *backoff* en los modelos de n-gramas bilingües como si de un modelo no-determinista se tratara, condicionando las transiciones en niveles inferiores en función de las que se hayan producido superiormente. Nuevamente, existe un sistema *baseline* para este apartado, que corresponde a las columnas de las tablas etiquetadas como "Suavizado estándar". En este caso, una arista de *backoff* se considera simplemente una transición de fallo, lo que impide explorar muchas opciones de análisis en la etapa de búsqueda.

Para todas las configuraciones estudiadas se muestra el efecto del parámetro n, el orden del modelo de n-gramas usado, para valores de $n = \{1, \ldots, 5\}$.

El resultado de los experimentos realizados sobre el corpus del EuroParl se muestra entre las tablas A.1 y A.5, ambas inclusive. Cada tabla hace referencia a la versión concreta del corpus y al sentido de traducción practicado. La aplicación de una integración log-lineal de modelos se expresa en cursiva. Los mejores resultados de cada tabla se resaltan mediante el formato negrita. En caso de igualdad, se marca el correspondiente al modelo de menor orden.

Tabla A.1: Resultados para el corpus francés→inglés del EuroParl 2006

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	4.9	72.2	72.7	4.9	72.2	72.7
2	18.8	64.9	62.6	18.9	64.6	62.3
3	19.9	64.0	62.1	20.4	63.4	61.5
4	20.2	64.1	62.2	21.2	63.1	61.0
5	19.8	64.4	62.6	21.2	63.0	60.9
	Segment. mediante aglutinamiento mínimo					
1	10.5	71.4	69.0	10.5	71.4	69.0
2	24.8	64.8	60.6	25.6	63.8	59.5
3	25.1	65.3	61.1	26.8	63.2	58.9
4	25.1	65.3	61.1	27.0	63.2	58.9
	—	—	—	*28.4*	*60.2*	*55.9*
5	25.1	65.3	61.1	26.9	63.3	59.0
	Segmentación usando Pharaoh					
1	26.8	62.3	57.8	26.8	62.3	57.8
2	26.3	63.9	59.4	28.0	61.9	57.3
3	25.8	64.5	60.0	27.9	62.0	57.4
4	25.7	64.5	60.0	27.9	62.0	57.4
5	25.7	64.5	60.0	27.9	62.0	57.4
	Segmentación usando Moses					
1	28.2	59.0	54.8	28.2	59.0	54.8
2	28.2	59.8	55.4	**29.3**	**58.9**	**54.5**
3	27.9	60.1	55.7	29.2	58.9	54.5
4	28.0	60.1	55.6	29.2	58.9	54.5
5	27.9	60.1	55.6	29.2	58.9	54.5

Tabla A.2: Resultados para el corpus español→inglés del EuroParl 2006

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	5.9	71.4	71.5	5.9	71.4	71.5
2	20.3	64.1	61.9	20.4	63.7	61.5
3	20.5	63.5	61.8	21.3	62.5	60.7
4	20.3	63.6	61.9	21.3	62.5	60.7
5	20.1	63.8	62.1	21.4	62.4	60.4
	Segment. mediante aglutinamiento mínimo					
1	12.1	69.7	67.1	12.1	69.7	12.1
2	25.4	63.8	59.5	25.9	62.9	58.5
3	25.5	64.4	60.2	26.8	62.8	58.3
4	25.4	64.5	60.2	27.0	62.8	58.4
5	25.4	64.5	60.2	27.0	62.8	58.5
	Segmentación usando Pharaoh					
1	26.2	62.2	57.6	26.2	62.2	57.6
2	26.0	63.2	58.5	27.5	**61.6**	**57.1**
3	25.5	63.7	59.1	**27.6**	61.7	57.1
4	25.5	63.8	59.1	27.5	61.7	57.2
5	25.5	63.7	59.1	27.5	61.7	57.2

Tabla A.3: Resultados para el corpus inglés→español del EuroParl 2006

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	2.6	75.5	77.0	2.6	75.5	77.0
2	16.8	67.9	66.4	16.8	67.6	66.1
3	15.6	69.0	68.3	16.8	67.5	66.7
4	14.9	69.7	69.1	16.6	67.6	66.8
5	14.5	70.0	69.5	16.6	67.7	66.9
	Segment. mediante aglutinamiento mínimo					
1	10.9	72.1	69.4	10.9	72.1	69.4
2	24.6	64.2	60.1	24.8	63.5	59.3
3	25.3	64.5	60.3	26.0	63.3	59.1
4	25.3	64.4	60.2	26.2	63.2	59.1
5	25.2	64.4	60.2	26.1	63.3	59.1
	Segmentación usando Pharaoh					
1	24.9	63.1	58.7	24.9	63.1	58.7
2	25.4	63.8	59.2	**26.4**	**62.3**	**57.7**
3	24.8	64.3	59.7	26.2	62.4	57.9
4	24.8	64.3	59.7	26.2	62.4	57.9
5	24.8	64.3	59.7	26.2	62.4	57.9

Tabla A.4: Resultados para el corpus español→inglés del EuroParl 2007

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	7.6	68.6	68.0	7.6	68.6	68.0
2	21.9	62.9	60.5	22.0	62.6	60.2
3	22.1	62.1	60.1	23.0	61.2	59.1
4	21.8	62.5	60.7	23.0	61.3	59.2
5	21.7	62.7	60.8	23.0	61.2	59.1
	Segment. mediante aglutinamiento mínimo					
1	12.9	68.5	65.7	12.9	68.5	65.7
2	26.1	62.8	58.5	26.7	62.0	57.6
3	26.3	63.2	58.9	27.8	61.6	57.3
4	26.3	63.2	59.0	27.9	61.6	57.2
5	26.2	63.3	59.0	27.9	61.7	57.4
	Segmentación usando Pharaoh					
1	26.4	59.7	55.2	26.4	59.7	55.2
2	26.7	60.8	56.2	27.9	59.5	54.9
3	26.3	61.2	56.6	28.0	59.5	54.9
4	26.2	61.2	56.6	**28.1**	**59.5**	**54.9**
5	26.2	61.2	56.7	28.1	59.6	55.0

Tabla A.5: Resultados para el corpus inglés→español del EuroParl 2007

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	3.9	73.0	73.5	3.9	73.0	73.5
2	20.1	64.9	62.6	20.0	64.6	62.4
3	19.4	65.5	63.9	20.3	64.3	62.6
4	18.8	66.0	64.7	20.1	64.4	62.8
5	18.7	66.2	64.9	20.4	64.2	62.5
	Segment. mediante aglutinamiento mínimo					
1	11.0	71.4	68.6	11.0	71.4	68.6
2	25.2	63.4	59.4	25.4	62.8	58.6
3	26.0	63.8	59.7	**26.8**	62.6	58.4
4	25.7	63.9	59.8	26.8	62.6	58.4
5	25.6	64.0	59.9	26.7	62.6	58.4
	Segmentación usando Pharaoh					
1	24.1	61.3	57.2	24.1	61.3	57.2
2	24.2	62.1	57.9	25.2	**60.8**	**56.6**
3	23.8	62.4	58.2	25.2	60.9	56.7
4	23.8	62.4	58.2	25.2	60.9	56.7
5	23.8	62.4	58.2	25.2	60.9	56.7

Los resultados con la tarea de Xerox se presentan de la tabla A.6 a la A.11. Se agrupan por pares, reflejando los dos sentidos de traducción por partición. La segmentación basada en segmentos es mediante aglutinamiento mínimo.

Tabla A.6: Resultados para el corpus inglés→alemán de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	1.4	80.2	80.4	1.4	80.2	80.4
2	16.5	73.3	72.1	16.3	73.4	72.1
3	17.7	71.7	70.3	18.2	70.9	69.6
4	18.0	71.3	69.8	18.6	70.7	69.2
5	18.1	71.4	69.9	18.6	70.8	69.2
	Segmentación basada en segmentos					
1	16.8	74.4	72.7	16.8	74.4	72.7
2	22.6	70.1	68.1	22.9	69.4	67.6
3	22.0	71.0	69.2	23.0	69.3	67.6
4	22.2	70.9	69.0	**23.0**	**69.2**	**67.5**
5	22.2	70.8	69.0	23.0	69.3	67.5

Tabla A.7: Resultados para el corpus alemán→inglés de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	5.5	75.2	74.7	5.5	75.2	74.7
2	23.8	64.8	63.2	24.2	64.6	62.9
3	24.7	63.8	62.3	25.4	63.2	61.6
4	24.5	64.1	62.5	25.2	63.4	61.7
5	24.6	64.1	62.5	25.3	63.3	61.7
	Segmentación basada en segmentos					
1	20.8	62.3	59.8	20.8	62.3	59.8
2	29.3	59.0	56.2	29.5	58.7	55.9
3	29.7	58.7	55.8	**30.2**	**58.2**	**55.3**
4	29.7	58.8	55.8	30.1	58.3	55.4
5	29.7	58.8	55.8	30.2	58.2	55.4

Tabla A.8: Resultados para el corpus inglés→español de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	13.0	61.3	60.6	13.0	61.3	60.6
2	51.5	37.1	35.3	51.0	37.4	35.6
3	57.7	32.2	30.4	57.9	31.7	30.0
4	58.0	32.1	30.4	58.3	31.5	29.8
5	58.3	32.1	30.3	58.3	31.6	29.9
	Segmentación basada en segmentos					
1	38.3	40.8	39.2	38.3	40.8	39.2
2	61.0	28.5	26.4	61.3	28.1	25.9
3	60.8	28.8	26.6	**62.3**	**27.5**	**25.3**
4	60.5	28.9	26.7	62.1	27.6	25.4
5	60.5	28.9	26.7	62.1	27.6	25.4

Tabla A.9: Resultados para el corpus español→inglés de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	16.5	57.1	56.5	16.5	57.1	56.5
2	39.1	43.9	42.7	39.2	43.7	42.5
3	49.2	35.4	34.4	49.5	35.0	33.9
4	51.6	34.0	33.0	52.2	33.5	32.4
5	52.1	33.8	32.7	52.5	33.3	32.2
	Segmentación basada en segmentos					
1	43.0	36.2	34.2	43.0	36.2	34.2
2	56.5	29.9	27.6	57.7	28.7	26.6
3	56.3	30.3	28.1	58.9	28.1	25.9
4	56.4	30.2	28.0	**58.9**	**28.0**	**25.8**
5	56.3	30.3	28.1	58.7	28.1	25.9

Tabla A.10: Resultados para el corpus inglés→francés de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	9.4	71.2	71.0	9.4	71.2	71.0
2	25.5	64.7	62.8	25.6	64.7	62.8
3	27.5	63.3	61.3	28.0	62.8	61.0
4	27.9	62.6	60.9	28.3	62.4	60.5
5	27.7	62.4	60.6	28.2	62.0	60.2
	Segmentación basada en segmentos					
1	19.2	64.9	62.8	19.2	64.9	62.8
2	32.6	58.6	55.8	32.9	58.2	55.4
3	33.5	58.1	55.4	**34.4**	**57.7**	**54.9**
4	33.4	58.2	55.5	34.4	57.8	55.1
5	33.5	58.2	55.5	34.2	57.8	55.1

Tabla A.11: Resultados para el corpus francés→inglés de Xerox

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	17.1	60.9	59.9	17.1	60.9	59.9
2	27.8	59.5	57.8	27.6	59.7	57.9
3	29.4	56.8	55.3	29.6	57.1	55.6
4	28.7	57.9	56.2	30.2	56.4	54.7
5	29.1	57.3	55.6	29.5	56.9	55.1
	Segmentación basada en segmentos					
1	20.9	65.2	62.3	20.9	65.2	62.3
2	30.3	58.5	55.8	30.6	58.2	55.4
3	31.7	57.7	54.8	32.5	57.4	54.5
4	31.3	58.1	55.2	32.6	57.6	54.7
	—	—	—	***34.2***	***52.3***	***49.2***
5	31.2	58.2	55.3	32.1	57.8	54.9

Los resultados mediante el corpus i3media se presentan en la tabla A.12. La segmentación basada en segmentos es mediante aglutinamiento mínimo.

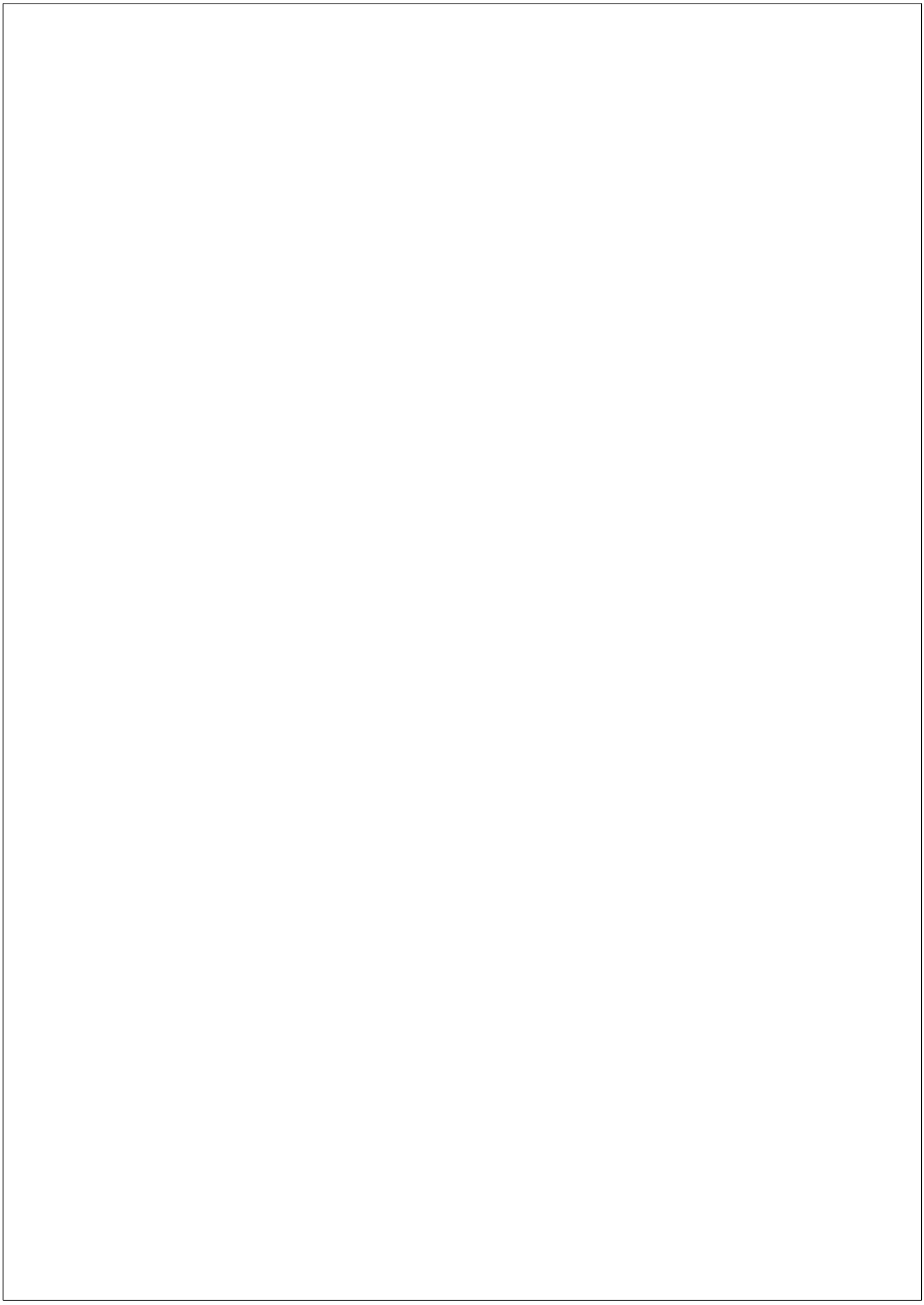
Tabla A.12: Resultados para el corpus español→catalán de i3media

	Suavizado estándar			*Backoff* condicional		
n	BLEU	WER	TER	BLEU	WER	TER
	Segmentación basada en palabras					
1	66.9	18.9	18.4	66.9	18.9	18.4
2	79.9	13.4	13.0	80.8	12.9	12.5
3	79.5	12.7	12.3	82.3	11.1	10.7
4	79.1	12.9	12.5	82.6	10.8	10.5
5	78.9	13.0	12.6	82.6	10.9	10.5
	Segmentación basada en segmentos					
1	58.5	24.2	23.7	58.5	24.2	23.7
2	80.3	12.5	12.1	81.4	11.7	11.3
3	80.5	12.2	11.8	83.1	10.6	10.2
4	80.2	12.3	11.9	83.4	10.3	9.9
5	80.1	12.3	11.9	83.5	10.2	9.9
	—	—	—	***83.9***	***9.9***	***9.5***

A continuación, la tabla A.13 muestra los resultados de traducción sobre el corpus EuTrans mediante el esquema basado en categorías del Capítulo 5.

Tabla A.13: Resumen de resultados de traducción sobre el corpus EuTrans. Se comparan tres sistemas diferentes: a) el que se deriva del corpus original; b) aplicando un preproceso basado en técnicas lingüísticas; c) utilizando el marco de categorización propuesto a través de la morfología de las palabras.

	Tasas	
Sistema	BLEU	WER
Baseline	88.0	8.3
Preproceso	88.0	**8.1**
Categorización	78.9	13.1



Bibliografía

[ABC+00] J.C. Amengual, J.M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[ACC+06] Jordi Atserias, Bernardino Casas, Elisabet Comelles, Meritxell González, Lluís Padró, and Muntsa Padró. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May 2006. ELRA.

[AGGM06] Enrique Amigó, Jesús Giménez, Julio Gonzalo, and Lluís Màrquez. Mt evaluation: human-like vs. human acceptable. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 17–24, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[AH07] Joshua Albrecht and Rebecca Hwa. Regression for sentence-level mt evaluation with pseudo references. In *ACL*, pages 296–303. The Association for Computer Linguistics, 2007.

[AM04] Cyril Allauzen and Mehryar Mohri. An optimal pre-determinization algorithm for weighted transducers. *Theoretical Computer Science*, 328(1–2):3–18, November 2004.

[AOCBF+07] Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Marco A. Montava Belda, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, and Felipe Sánchez-Martínez. Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. In J. Rafael Rodríguez Galván and Ma-

nuel Palomo Duarte, editors, *Proceedings of the FLOSS International Conference 2007*, pages 5–20. Servicio de Publicaciones de la Universidad de Cadiz, 2007.

[AU72] A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, 1972.

[BBC+09] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda H. Ney, J. Tomás, and E. Vidal. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.

[BBDP+96] Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Andrew S. Kehler, and Robert L. Mercer. Language translation apparatus and method using context-based translation models, April 1996.

[BCDP+90] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roosin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[BDPDP96] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[Ber79] J. Berstel. *Transductions and context-free languages*. B.G.Teubner Stuttgart, 1979.

[Bor02] Lars Borin, editor. *Parallel corpora, parallel worlds.* Language and Computers Vol. 43. Rodopi, Kenilworth, 2002.

[BPPM93] Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[BR01] S. Bangalore and G. Riccardi. A finite-state approach to machine translation. In *Proceedings of the NAACL*, 2001.

[BR03] S. Bangalore and G. Riccardi. Stochastic finite-state models for spoken language machine translation. *Machine Translation*, 17(3):165–184, 2003.

[Cas90] F. Casacuberta. Some relations among stochastic finite state networks used in automatic speech recognition. *IEEE Trans. on PAMI*, 12(7):691–695, 1990.

[Cas95] F. Casacuberta. Probabilistic estimation of stochastic regular syntax-directed translation schemes. In A. Calvo and R. Medina, editors, *VI Spanish Symposium on Pattern Recognition and Image Analysis*, pages 201–207, Córdoba, España, 1995. AERFAI.

[Cas00] F. Casacuberta. Inference of finite-state transducers by using regular grammars and morphisms. In A.L. Oliveira, editor, *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 2000. 5th International Colloquium Grammatical Inference (ICGI2000). Lisbon. Portugal. September.

[CBFK+07] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. (Meta-) Evaluation of Machine Translation. In *Proc. of the 2nd Workshop on SMT*, pages 136–158, Prague, Czech Republic, 2007. ACL.

[CBFK+08] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[CBOK06] C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the Role of Bleu in Machine Translation Research. In *Proceedings the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italia, 2006.

[CCC+08] 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 ACM*, page In press, 2008.

[CCM+94] S. Campos, E. Clarke, W. Marrero, M. Minea, and H. Hiraishi. Computing quantitative characteristics of finite state real-time systems. In *IEEE Real-Time Systems Symposium*, pages 266–270. IEEE, 1994.

[CdlH00] F. Casacuberta and C. de la Higuera. Computational complexity of problems on probabilistic grammars and transducers. In *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 15–24. Springer-Verlag, 2000.

[CG96] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In Aravind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318. Morgan Kaufmann Publishers, 1996.

[Chi05] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Chi07] David Chiang. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, 2007.

[CNO+04] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, S. Molau, F. Nevado, M. Pastor, D. Picó, and A. Sanchis. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[CO94] R. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *Proc. 2nd International Colloquium on Grammatical Inference - ICGI '94*, volume 862, pages 139–150. Springer-Verlag, 1994.

[CV04] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[CVP05] Francisco Casacuberta, Enrique Vidal, and David Picó. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38(9):1431–1443, September 2005.

[DGM06] A. De Gispert and J. B. Mariño. Linguistic knowledge in statistical phrase-based word alignment. *Nat. Lang. Eng.*, 12(1):91–108, 2006.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[Dod02] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-ocurrence statistics. In *Human Language Technology Workshop*, pages 128–132. ARPA, 2002.

[DR97] Pierre Dupont and Ronald Rosenfeld. Lattice based language models. Technical Report CMU-CS-97-173, Carnegie Mellon University, 1997.

[DSE08] Markus Dreyer, Jason R. Smith, and Jason Eisner. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1080–1089, Honolulu, October 2008.

[Eis02] Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–8, Philadelphia, July 2002.

[ELVL04] José Esteban, José Lorenzo, Antonio S. Valderrábanos, and Guy Lapalme. Transtype2 - an innovative computer-assisted translation system. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 94–97, Barcelona, Spain, jul 2004. Association for Computational Linguistics. TT2.

[For73] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[Fu82] K. S. Fu. *Syntactic pattern recognition and applications*. Prentice-Hall, Englewood Cliffs, NJ., 1982.

[GC07] J. González and F. Casacuberta. Phrase-based finite state models. In *Proceedings of the 6th International Workshop on Finite State Methods and Natural Language Processing (FSMNLP)*, Potsdam (Germany), September 14-16 2007.

[Ger03] U. Germann. Greedy decoding for statistical machine translation in almost linear time. Proceedings of HLT-NAACL. Edmonton, AB, Canada, 2003.

[GJK+01] Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 228–235, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[GLN+06] J. González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, A. Giménez, F. Casacuberta, J. M de Val, and F. Fabregat. SisHiTra: a Spanish-to-Catalan hybrid machine translation system. In *5th SALTMIL Workshop on Minority Languages*, pages 69–73, Genoa, Italy, 23 May 2006.

[GM08] Jesús Giménez and Lluís Màrquez. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation*, pages 195–198, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[Goo03] Joshua Goodman. The state of the art in language modeling. In *Proceedings of HLT-NAACL*, Edmonton, 2003.

[GSC08] J. González, G. Sanchis, and F. Casacuberta. Learning finite state transducers using bilingual phrases. In *9th International Conference on Intelligent Text Processing and Computational Linguistics. Lecture Notes in Computer Science*, Haifa, Israel, February 17 to 23 2008.

[HKPB02] Eduard Hovy, Margaret King, and Andrei Popescu-Belis. Principles of context-based machine translation evaluation. *Machine Translation*, 17(1):43–75, 2002.

[HMU06] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 2006.

[HS78] Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, 1978.

[HS09] Martin Haspelmath and Andrea Sims. *Understanding Morphology*. Understanding Language Series. Hodder Education, London, 2nd edition, 2009.

[IC97] Pierre Isabelle and Kenneth Ward Church. *Machine Translation*, 12(1-2), 1997.

[Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[JM80] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands, 1980.

[JM08] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall in Artificial Intelligence, 2nd edition, May 2008.

[KAO98] K. Knight and Y. Al-Onaizan. Translation with finite-state devices. In *Proceedings of the 4th. ANSTA Conference*, 1998.

[Kat87] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Translations on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[KB03] Shankar Kumar and William Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 63–70, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[KB05] Shankar Kumar and William Byrne. Local phrase reordering models for statistical machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 161–168, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[KDB06] Shankar Kumar, Yonggang Deng, and William Byrne. A weighted finite state transducer translation template model for statistical machine translation. *Nat. Lang. Eng.*, 12(1):35–75, 2006.

[KH07] Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[KHB+07] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics, 2007.

[KM06] Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings on the HTL-NAACL Workshop on Statistical Machine Translation*, pages 102–121. Association for Computational Linguistics, 2006.

[KN95] Reinhard Kneser and Hermann Ney. Improved backing-off for *n*-gram language modeling. In *International Conference on Acoustic, Speech and Signal Processing*, pages 181–184, 1995.

[Kne96] Reinhard Kneser. Statistical language modeling using a variable context length. In *Proceedings of ICSLP*, volume 1, pages 494–497, Philadelphia, October 1996.

[Koe04] Philipp Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124, 2004.

[Koe05] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, 2005.

[KOM03] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Kos83] Kimmo Koskenniemi. Two-level morphology: A general computational model of word-form recognition and production. Technical Report Publication No. 11, Department of General Linguistics, University of Helsinki, 1983.

[KPBH03] M. King, A. Popescu-Belis, and E. Hovy. FEMTI: creating and using a framework for MT evaluation. In *Machine Translation Summit IX*, pages 224–231, New Orleans, USA, September 2003.

[KS04] Alex Kulesza and Stuart M. Shieber. A learning approach to improving sentence-level mt evaluation. In *In Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI*, 2004.

[KVM+05] Stephan Kanthak, David Vilar, Evgeny Matusov, Richard Zens, and Hermann Ney. Novel reordering approaches in phrase-based statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 167–174, 2005.

[LA07] Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[LG07] Ding Liu and Daniel Gildea. Source-language features and maximum correlation training for machine translation evaluation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 41–48, Rochester, New York, April 2007. Association for Computational Linguistics.

[Llo00] D. Llorens. *Suavizado de autómatas y traductores finitos estocásticos*. PhD thesis, Universitat Politècnica de València, 2000.

[Lop08] Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3), 2008.

[LRL05] Lucian Lita, Monica Rogati, and Alon Lavie. Blanc: Learning evaluation metrics for mt. In *Proceedings of Human Language*

Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 740–747, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[Mäk99] E. Mäkinen. Inferring finite transducers. Technical Report A-1999-3, University of Tampere, 1999.

[Mat92] Peter Matthews. *Morphology*. Cambridge University Press, 2nd edition, 1992.

[MBC+06] José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. N-gram-based machine translation. *Comput. Linguist.*, 32(4):527–549, 2006.

[MGT03] I. Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of HLT/NAACL*, pages 61–63, 2003.

[Moh94] Mehryar Mohri. Minimization of sequential transducers. In *Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching (CPM '94)*, volume 807 of *Lecture Notes in Computer Science*, pages 151–163, Asilomar, California, June 1994. Springer-Verlag.

[MPR02] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

[MW02] Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 133–139, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[NEK94] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language*, 8:1–38, 1994.

[NM65] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[NMNP92] H. Ney, D. Mergel, A. Noll, and A. Paeseler. Data driven search organization for continuous speech recognition. *IEEE Transactions on Signal Processing*, 40(2):272–281, February 1992.

[NN04] Sonja Nießen and Hermann Ney. Statistical machine translation with scarce resources using morpho-syntactic information. *Comput. Linguist.*, 30(2):181–204, 2004.

[NNO+00] H. Ney, S. Nießen, F. Och, H. Sawaf, C. Tillmann, and S. Vogel. Algorithms for statistical translation of spoken language. *IEEE Transactions on Speech and Audio Processing*, 8(1):24–36, 2000.

[NOLN00] S. Nießen, F. J. Och, G. Leusch, and H. Ney. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of 2nd International Conference on Language Resources and Evaluation (LREC)*, Athens, Greece, May 2000.

[OGV93] J. Oncina, P. Garcia, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458, 1993.

[OGVC03] D. Ortiz, I. García-Varea, and F. Casacuberta. An empirical comparison of stack-decoding algorithms for statistical machine translation. In *Pattern Recongnition and Image Analysis, First Iberia Conference*, volume 2652 of *Lecture Notes in Computer Science*, pages 654–663. Springer-Verlag, Puerto de Andratx, Mallorca, June 2003.

[OGVC05] D. Ortiz, I. García-Varea, and F. Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Tenth Machine Translation Summit*, pages 141–148. Asia-Pacific Association for Machine Translation, Phuket, Thailand, September 2005.

[ON03] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March 2003.

[ON04] Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, 2004.

[OTN99] F. J. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99) in conjunction with the 37th annual meeting of the Association for Computational Linguistics*, pages 20–28, Washington, MD, Jun 1999.

[Paz71] Azaria Paz. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press, Inc., Orlando, FL, USA, 1971.

[PC01] D. Picó and F. Casacuberta. Some statistical-estimation methods for stochastic finite-state transducers. *Machine Learning*, 44(1-2):121–141, 2001.

[PCTG05] A. Pérez, F. Casacuberta, M. I. Torres, and V. Guijarrubia. Finite-state transducers based on k-tss grammars for speech translation. In *Proceedings of Finite-State Methods and Natural Language Processing (FSMNLP 2005)*, pages 270–272, Helsinki, Finland, September 2005.

[PFS07] Michael Paul, Andrew Finch, and Eiichiro Sumita. Reducing human assessment of machine translation quality to binary classifiers. In *Conference on Theoretical and Methodological Issues in Machine Translation*, pages 154–162, Skövde, Sweden, September 2007.

[PGCT04] A. Pérez, J. González, F. Casacuberta, and M. I. Torres. Traducción automática mediante transductores estocásticos de estados finitos. In *Actas de las Terceras Jornadas de Tecnología del Habla*, Valencia, Spain, November 2004.

[PGJ+07] A. Pérez, V. Guijarrubia, R. Justo, M. I. Torres, and F. Casacuberta. A comparison of linguistically and statistically enhanced models for speech-to-speech machine translation. In *Proceedings of the 4th International Workshop on Spoken Language Translation*, pages 13–20, Trento, Italy, October 15–16 2007.

[Pic05] David Picó. *Combining Statistical and Finite-State Methods for Machine Translation*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia (Spain), September 2005. Advisor: Dr. F. Casacuberta.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Computational Linguistics (ACL), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia*, pages 311–318, 2002.

[PTC06] Alicia Pérez, Inés Torres, and Francisco Casacuberta. Towards the improvement of statistical translation models using linguistic features. In *Advances in Natural Language Processing, Proceedings of the 5th International Conference on NLP, FinTAL*, volume 4139 of *Lecture Notes in Computer Science*, pages 716–725. Springer, Turku, Finland, August 23-25 2006.

[PTC07] A. Pérez, M. I. Torres, and F. Casacuberta. Speech translation with phrase based stochastic finite-state transducers. In *Proceedings of the 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, volume IV, pages 113–116, Honolulu, Hawaii USA, April 15-20 2007. IEEE.

[PTC08] A. Pérez, M. I. Torres, and F. Casacuberta. Joining linguistic and statistical methods for Spanish-to-Basque speech translation. *Speech Communication*, 50:1021–1033, 2008.

[PTCG06] A. Pérez, M. I. Torres, F. Casacuberta, and V. Guijarrubia. A Spanish-Basque weather forecast corpus for probabilistic speech translation. In *5th SALTMIL Workshop on Minority Languages*, pages 99–102, Genoa, Italy, 23 May 2006.

[Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[RJ86] L. Rabiner and B. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, 3(1):4–16, 1986.

[RJ93] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[Ros96] Ronald Rosenfeld. A maximum entropy approach to adaptative statistical language modeling. *Computer Speech and Language*, (10):187–228, 1996.

[Ros00] R. Rosenfeld. Two decades of statistical language modelling: Where do we go from here? In *Proceedings of the IEEE*, volume 88 (8), pages 1270–1278, 2000.

[SB99] Masayoshi Shibatani and Theodora Bynon, editors. *Approaches to Language Typology*. Oxford University Press, March 1999.

[SDS+06] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings AMTA*, pages 223–231, August 2006.

[SH07] Anil Kumar Singh and Samar Husain. Exploring translation similarities for building a better sentence aligner. In *Proceedings of the 3rd Indian International Conference on Artificial Intelligence*, pages 1852–1863, Pune, India, 2007.

[STN94] Volker Steinbiss, Bach-Hiep Tran, and Hermann Ney. Improvements in beam search. In *Proc. of 3rd International Conference on Spoken Language Processing (ICSLP'94)*, pages 2143–2146, Yokohama (Japan), September 1994.

[Stu01] Gregory T. Stump. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge University Press, United Kingdom, 2001.

[TC01] J. Tomás and F. Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the Machine Translation Summit VIII*, pages 357–361, Santiago de Compostela, 2001.

[TCS07] TC-STAR project: Technology and corpora for Speech to Speech translation. http://www.tc-star.org, 2004-2007.

[TdVF+01] J. Tomás, J. M. de Val, F. Fabregat, F. Casacuberta, D. Picó, A. Sanchis, and E. Vidal. Automatic development of spanish-catalan corpora for machine translation. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies*, pages 175–179, Jaén, 2001.

[TN03] Christoph Tillmann and Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.*, 29(1):97–133, 2003.

[Tom03] Jesús Tomás. *Traducción automática de textos entre lenguas similares utilizando métodos estadísticos*. PhD thesis, Universidad Politécnica de Valencia, Valencia (Spain), July 2003. (In Spanish).

[VGS89] E. Vidal, P. García, and E. Segarra. Inductive learning of finite-state transducers for the interpretation of unidimensional objects. In R. Mohr, Th. Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, pages 17–35. World Scientific pub, 1989.

[Vid97] E. Vidal. Finite-state speech-to-speech translation. In *Proc. of the International Conference on Acoustic Speech and Signal Processing*, pages 111–114. IEEE, 1997.

[Vil00] J. M. Vilar. Improving the learning of subsequential transducers by using alignments and dictionaries. In *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Artificial Intelligence*, pages 298–312. Springer-Verlag, 2000.

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, pages 260–269, 1967.

[WAC03] Y. Wang, A. Acero, and C. Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Virgin Islands, 2003.

[Wet80] C. S. Wetherell. Probabilistic languages: a review and some open questions. *Computing Surveys*, 12(4):361–379, 1980.

[WOO94] John S. White, T. O'Connell, and F. O'Mara. The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Further Approaches. In *Proceedings of the 1994 Conference of the Association for Machine Translation in the Americas*, pages 193–205, 1994.

[ZLH06] Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 77–84, Sydney, Australia, July 2006. Association for Computational Linguistics.

[ZON02] Richard Zens, Franz Josef Och, and Hermann Ney. Phrase-based statistical machine translation. In *Proceedings of the 25th Annual German Conference on AI*, pages 18–32. Springer-Verlag, 2002.