UNIVERSIDAD POLITÉCNICA DE VALENCIA

UNIVERSIDAD
POLITECNICA
DE VALENCIA

Departamento de Sistemas Informáticos y Computación

Research report:

MACHINE TRANSLATION:
NEW APPROACHES TO PARAMETER OPTIMIZATION AND ALIGNMENT

Presented by:

Jesús González Rubio

Supervisors:

Francisco Casacuberta Nolla
Alfons Juan Ciscar

September 2008

Acknowledgements

This work has been partially supported by the Spanish MEC under FPU scholarship AP2006-00691 and grant Consolider Ingenio 2010 CSD2007-00018 and by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01.

Contents

List of Tables

CHAPTER 1

INTRODUCTION

Machine Translation (MT) is a research field of great importance in the European Community, where language plurality implies both a very important cultural richness and not negligible obstacle towards building a unified Europe. Because of this, a growing interest on MT has been shown both by politicians and research groups, which become more and more specialized in this field. In addition, Statistical Machine Translation (SMT) systems have proved in the last years to be an important alternative to rule-based MT systems, even outperforming commercial MT systems in the tasks they have been trained on. Moreover, the development effort behind a rule-based MT system and an SMT system is dramatically different, the latter being able to adapt to new language pairs with little or no human effort, whenever suitable corpora are available.

The goal of MT is the translation of a text given in some source language into a target language. We are given a source language sentence $\mathbf{f} = f_1 \dots f_j \dots f_J$ which is to be translated into a target language sentence. Among all possible target language sentences, we will choose the sentence $\hat{\mathbf{e}} = e_1 \dots e_i \dots e_I$ which maximizes the posterior probability. Such statement is formalized in the Fundamental Equation of Machine Translation:

$$\hat{\mathbf{e}} \;=\; \underset{\mathbf{e}}{\operatorname{argmax}}\{Pr(\mathbf{e}|\mathbf{f})\} \;=\; \underset{\mathbf{e}}{\operatorname{argmax}}\{Pr(\mathbf{e}) \cdot Pr(\mathbf{f}|\mathbf{e})\}\,. \tag{1.1}$$

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. The decomposition in Eq. (1.1) allows an independent modelling of the target *language model* $Pr(\mathbf{e})$ and the (inverse) *translation model* $Pr(\mathbf{f}|\mathbf{e})$[1], known as source-channel model [1]. This decomposition has a very intuitive interpretation: the translation model $Pr(\mathbf{f}|\mathbf{e})$ will capture the word relations between both input and output languages, whereas the language model $Pr(\mathbf{e})$ will ensure that the output sentence is a well-formed sentence belonging to the target language.

Word-based translation models were later on extended by phrase-based models [2, 3, 4], which have proved to provide a very efficient framework for machine translation. Phrase-based models compute the translation probability of a given *phrase*, i.e. sequence of words, and hence they introduce information about context. SMT systems implementing these models have mostly outperformed single-word models such as IBM Model 1 [5], becoming predominant in the state-of-the-art [6] nowadays.

[1] We use $Pr(\cdot)$ to denote general probability distributions and $p(\cdot)$ to denote model-based probability distributions.

1.1 SMT Alignment Models

Many SMT models [5, 7, 8, 9] try to model word-to-word correspondences between source and target words. Known as statistical alignment models, these models typically yield the following equation:

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} \{Pr(\mathbf{a}|\mathbf{e}) \cdot Pr(\mathbf{f}|\mathbf{e},\mathbf{a})\} \; . \tag{1.2}$$

The alignment model in Eq. (1.2) models the relations between the words of the input and the output sentences by introducing a 'hidden' word alignment $\mathbf{a} = a_1 \dots a_j \dots a_J$ into the translation model formulation. This alignment describes a mapping from a position j in the source sentence to a position a_j in the target sentence.

IBM Model 1 [5], is a word alignment model which was originally developed to provide reasonable initial parameter estimates for more complex word alignment models, but it has subsequently found a host of additional uses, as segmenting long sentences for improved word alignment [10] or extracting parallel sentences from comparable corpora [11]. Furthermore, at the 2003 John Hopkins summer workshop on statistical machine translation, a large number of features were tested to discover which ones could improve a state-of-the-art translation system, and the only feature that produced a "truly significant improvement" was the IBM Model 1 score [12].

IBM Model 1 is defined as a particularly simple alignment model, where all word-to-word alignments have the same probability, i.e. $Pr(\mathbf{a}|\mathbf{e})$ is modelled using a uniform distribution (which [5] show yields Eq. (1.3)). Hence, word order does not affect alignment probabilities.

$$p(\mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} \left[\frac{1}{I+1} \sum_{i=0}^{I} p(f_j|e_i) \right] \; . \tag{1.3}$$

IBM Model 1 clearly has many shortcomings as a translation model due to its simplicity. The *distortion problem* and the fact that some words act as *garbage collectors* are some of them. The distortion problem is a structural limitation of the IBM Model 1 due to the fact that the position of any word in the target sentence is independent of the position of the corresponding word in the source sentence, or the positions of any other source language words or their translations. The other problem with IBM Model 1, as standardly trained, is that rare words in the source language tend to act as "garbage collectors" [13, 12], aligning too many words in the target sentence.

There are some proposals to reduce the shown problems of IBM Model 1: extend the word-to-word alignment approach allowing one-to-many alignments [14], or deal with problems related to the suboptimal performance of the standard training method for IBM Model 1 [15].

1.2 SMT Log-Linear Models

In order to combine the positive contributions of different approaches, SMT models can be merged using a log-linear combination [16]. Log-linear models are an approximation to the probability distribution $Pr(\mathbf{e}|\mathbf{f})$. In this framework, we have a set of M feature functions $h_m(\mathbf{e}|\mathbf{f}),\ m = 1, \dots, M$. For each feature function, there exists a model parameter $\omega_m,\ m = 1, \dots, M$. The following decision rule is obtained:

$$\hat{\mathbf{e}} \;=\; \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ \frac{\exp[\sum_{m=1}^{M} \omega_m h_m(\mathbf{e}|\mathbf{f})]}{\sum_{\mathbf{e}'} \exp[\sum_{m=1}^{M} \omega_m h_m(\mathbf{e}'|\mathbf{f})]} \right\} \;=\; \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \omega_m h_m(\mathbf{e}|\mathbf{f}) \right\} \; . \tag{1.4}$$

Selecting appropriate weights ω_m is essential in order to obtain good translation performance. In [17] the MERT algorithm was introduced. The MERT technique allows to find the values of the weights that minimize a

given error rate measure. This has become much more standard than optimizing the conditional probability of the training data given the model (i.e., a maximum likelihood criterion), as was common previously. In [17] was also stated that system performance is best when parameters are optimized using the same objective function that will be used for evaluation; BLEU [18], which computes the precision of unigrams, bigrams, trigrams and 4-grams[2] with respect to a reference translation, remains common for both purposes and is often retained for parameter optimization even when alternative evaluation measures [19, 20] are used.

The MERT technique relies on data sets in which source language sentences are paired with (sets of) reference translations. This technique applies an iterative and (locally) convergent strategy to find a set of weights which optimizes the BLEU score; a n-best list of translations provided by the decoder is exploited for this purpose after each translation step. At each iteration of the MERT procedure, the whole corpus is translated, and this process continues until convergence is reached.

The main disadvantage of the MERT procedure consists in its high time complexity. Such time complexity is due to the above mentioned iterative nature of the MERT procedure.

1.3 Support Vector Machines

Support Vector Machines (SVMs) are a learning method introduced by Vapnik in [21] and [22]. SVMs are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin, hence they are also known as maximum margin classifiers.

SVMs are well-founded in terms of computational learning theory and very open to theoretical understanding and analysis. In [23] a generalization of the multiclass SVM learning [24, 25] was introduced. Such a formulation involves features extracted jointly from inputs and outputs. The naive approach of treating each structure as a separate class is often unfeasible, since it leads to a multiclass problem with a very large number of classes. This problem is overcome by specifying discriminant functions that exploit the structure and dependencies within the outputs.

SVM^{struct}[3] [23] is a SVM algorithm for predicting multivariate or structured outputs. It performs supervised learning by approximating a mapping

$$H : \mathcal{X} \rightarrow \mathcal{Y} \,, \tag{1.5}$$

using labeled training examples $(x_1, y_1), \ldots, (x_n, y_n)$. However, unlike regular SVMs which consider only univariate predictions like in classification, SVM^{struct} can predict complex objects like trees, sequences, or sets. Examples of problems with complex outputs are natural language parsing, sequence alignment in protein homology detection, and Markov models for part-of-speech tagging. The SVM^{struct} algorithm can also be used for linear-time training of binary and multiclass SVMs under the linear kernel [26].

The 1-slack cutting-plane algorithm implemented in SVM^{struct} V3.00 uses a new but equivalent formulation of the structural SVM quadratic program which allows a cutting-plane algorithm that has time complexity linear in the number of training examples. The n-slack algorithm of SVM^{struct} V2.50 is described in [27, 23]. The SVM^{struct} implementation is based on the SVM^{light} quadratic optimizer [28].

SVM^{struct} can be thought of as an API for implementing different kinds of complex prediction algorithms, e.g. multiclass classification [23], label sequence learning [23], natural language parsing [23] and Protein Sequence Alignment [29].

[2] A n-grams is a sequence of n consecutive words.
[3] http://svmlight.joachims.org/svm_struct.html

1.4 Research Motivation

The aim of this research is to improve the translation process. On the one hand, we revisit the parameter optimization procedure, on the other hand, a new alignment model is presented.

Selecting an appropriate weight vector $\boldsymbol{\omega}$ for a log-linear combination is crucial to obtain a good translation quality. The MERT technique allows to find the values of the weights that minimize a given error rate measure. This algorithm proceeds iteratively translating the whole corpus at each iteration until convergence is reached. The main disadvantage of the MERT algorithm is its high computational costs, so, as an alternative, in Chapter 2 we propose a new technique based on SVMs. Specifically we use a generalization of the SVMs that allows to predict multivariate or structured outputs by exploiting the structure and dependencies between inputs and outputs. This procedure calculates the coefficients ω_m of a log-linear combination that minimize a desired error function. Our objective is to replace the slow iterative MERT algorithm by a new non-iterative algorithm based on a generalization of SVMs.

In order to model the probability distribution $Pr(\mathbf{e}|\mathbf{f})$ many SMT models try to model the correspondences between the words of the input and the output sentences. They are known as word alignment models. IBM Model 1 is one of the firsts and most widely used alignment models. Initially IBM Model 1 was designed to provide reasonable initial parameter estimates to more complex alignment models, but it has been used in a host of additional problems. IBM Model 1 is a very simple alignment model and has many disadvantages due to its simplicity. On Chapter 3 we present a new SMT alignment model intended to reduce some of the problems inherent to IBM Model 1. Our proposal extends IBM Model 1 by taking into account a given fixed segmentation of the source and target sentences in the estimation of the statistical dictionary. The complete mathematical description of this alignment model is available on Appendix A.

After revisiting the translation process presenting two possible improvements, on Chapter 4 we present a translation task involving dead languages. Specifically, we want to translate between Latin and Spanish. Chapter 4 describes the process to create a sentence aligned corpora and the experiments carried out using them. A list of websites containing resources related to the Latin–Spanish translation task is shown on Appendix B.

Finally, conclusions and a summary of the novel contributions of this research report are stated on Chapter 5.

CHAPTER 2

OPTIMIZATION OF LOG-LINEAR MACHINE TRANSLATION MODEL PARAMETERS USING SVMS

The aim of this research is to replace the slow iterative MERT procedure by a new non-iterative algorithm based on the SVM^{struct} algorithm. The proposed algorithm is able to perform the log-linear model parameter optimization with a linear time complexity. This chapter is organized as follows: first, Section 2.1 describes of how to adapt the SVM^{struct} to perform the optimization of parameter of a log-linear SMT model, next, Section 2.2 presents some details related to the implementation of our proposal, then, a description of the corpus used and the experimentation carried out is related in Sections 2.3 and 2.4, finally, conclusions and future work can be found on Section 2.5.

2.1 Structured SVMs for Log-linear Model Parameter Optimization

This section shows how to adapt the SVM^{struct} algorithm in order to perform the optimization of parameters of a log-linear SMT model. A log-linear model (see Section 1.2) implies the following decision rule:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ \sum_m \omega_m \theta_m(\mathbf{f}, \mathbf{e}) \right\} , \tag{2.1}$$

where θ_m are features of the hypothesis $\mathbf{e}$ and ω_m are weights associated with those features. The problem consists on selecting the appropriate vector of weights $\boldsymbol{\omega}$ so an objective function is optimized. SVMs are used to accomplish this optimization.

The vector $\boldsymbol{\omega}$ has a crucial influence on the quality of the translations. In the following, we aim to learn $\boldsymbol{\omega}$ from a set $\mathcal{T}$ of training examples:

$$\mathcal{T} = ((\mathbf{f}_1, \mathbf{e}_1), \ldots, (\mathbf{f}_n, \mathbf{e}_n), \ldots, (\mathbf{f}_N, \mathbf{e}_N)) , \tag{2.2}$$

where $(\mathbf{f}_n, \mathbf{e}_n)$ are sentence pairs.

This training set is assumed to be generated independently and identically distributed according to some unknown distribution $\mathcal{P}(F, E)$. A MT algorithm can be seen as a function:

$$h_{\boldsymbol{\omega}}(f) = \underset{\mathbf{e} \in E}{\operatorname{argmax}} \{\boldsymbol{\omega} \cdot \Psi(\mathbf{f}, \mathbf{e})\} , \tag{2.3}$$

which maps a given source sentence $\mathbf{f}$ to a target sentence $\mathbf{e}$. Our goal is to find a parameter vector $\boldsymbol{\omega}$ so that the predicted translation $h_{\boldsymbol{\omega}}(\mathbf{f})$ matches the correct translation on new test data as well as possible. In particular, we want to find the values of $\boldsymbol{\omega}$ that minimizes the expected loss (also called risk) for the data distribution $\mathcal{P}(F, E)$:

$$R_{\mathcal{P}}(h_{\boldsymbol{\omega}}) = \int \Delta(\mathbf{e}, h_{\boldsymbol{\omega}}(\mathbf{f}))d\mathcal{P}(F, E) \ , \tag{2.4}$$

where $\Delta(\mathbf{e}, \mathbf{e}')$ is a user defined (non-negative) loss function that quantifies how 'bad' it is to predict $\mathbf{e}'$ when $\mathbf{e}$ is the correct translation. For example, one may choose $\Delta(\mathbf{e}, \mathbf{e}')$ to be equal to 1 minus the BLEU score for $\mathbf{e}'$.

Following the principle of (Structural) Empirical Risk Minimization [30], finding a value of $\boldsymbol{\omega}$ that predicts well on new data can be achieved by minimizing the empirical loss (i.e the training error) on the training set $\mathcal{T}$.

$$R_{\mathcal{T}}(h_{\boldsymbol{\omega}}) = \sum_{n=1}^{N} \Delta(\mathbf{e}_n, h_{\boldsymbol{\omega}}(\mathbf{f}_n)) \ . \tag{2.5}$$

This minimization lead to the computational problem of finding the value of $\boldsymbol{\omega}$ which minimizes $R_{\mathcal{T}}(h_{\boldsymbol{\omega}})$. This vector $\boldsymbol{\omega}$ is the vector of optimized weights for the log-linear combination of models.

The problem of finding the value of $\boldsymbol{\omega}$ that minimizes the empirical loss $R_{\mathcal{T}}(h_{\boldsymbol{\omega}})$ of the translation algorithm was formulated as a minimization problem [23]:

$$\min_{\boldsymbol{\omega}, \xi} \left\{ \frac{1}{2}||\boldsymbol{\omega}||^2 + \frac{C}{n}\sum_{n=1}^{N} \xi_n \right\} \quad s.t \ \forall n \ : \ \xi_n \geq 0 \ ,$$

$$\forall n, \ \forall \mathbf{e} \in E \ : \ \boldsymbol{\omega} \cdot \delta\Psi_n(\mathbf{e}) \geq \Delta(\mathbf{e}_n, \mathbf{e}) - \xi_n \ , \tag{2.6}$$

where $\delta\Psi_n(\mathbf{e}) \ = \ \Psi(\mathbf{e}_n, \mathbf{f}_n) - \Psi(\mathbf{e}, \mathbf{f}_n)$.

The objective is the conventional regularized risk used in SVMs. The constraints in Eq. (2.6) state that the score $\boldsymbol{\omega} \cdot \Psi(\mathbf{e}_n, \mathbf{f}_n)$ of the correct translation $\mathbf{e}_n$ must be greater than the score $\boldsymbol{\omega} \cdot \Psi(\mathbf{e}, \mathbf{f}_n)$ of any other alternative translation $\mathbf{e}$.

This formulation includes a loss function $\Delta(\mathbf{e}_n, \mathbf{e})$ that scales the desired difference in score. Intuitively, the larger the loss of an alternative translation $\mathbf{e}$, the further should the score be away for that of the correct translation $\mathbf{e}_n$. ξ_n is a slack variable shared among constraints from the same example, since in general the constraint system is not feasible. In [31] is proved that this formulation minimizes training loss, while the SVM-style regularization with the norm $\boldsymbol{\omega}$ in the objective provides protection against over-fitting for high-dimensional $\boldsymbol{\omega}$. The parameter C allows the user to control the trade-off between training error and regularization.

The general training algorithm [23] can be seen in Figure 2.1. This algorithm requires the implementation of the feature mapping function $\Psi(\mathbf{f}, \mathbf{e})$, the loss function $\Delta(\mathbf{e}_n, \mathbf{e})$ and the maximization given in the 6^{th} line of the algorithm in order to be adapted to a specific task.

Following sections explain how to adapt the SVM^{struct} algorithm to perform the log-linear model parameter optimization.

2.1.1 Feature Mapping

The feature mapping function is a combined feature representation of the inputs and outputs. In our case, the mapping function takes a pair of input/output sentences and returns a vector with the scores of each of the models in the log-linear combination for this pair of sentences.

1: Input: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$, C, ϵ
2: $S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
3: **repeat**
4: **for** $i = 1, \ldots, n$ **do**
5: set up cost function
$\text{SVM}_1^{\triangle s}$: $H(\mathbf{y}) \equiv (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \triangle(\mathbf{y}_i, \mathbf{y})$
$\text{SVM}_2^{\triangle s}$: $H(\mathbf{y}) \equiv (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})}$
$\text{SVM}_1^{\triangle m}$: $H(\mathbf{y}) \equiv \triangle(\mathbf{y}_i, \mathbf{y}) - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle$
$\text{SVM}_2^{\triangle m}$: $H(\mathbf{y}) \equiv \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle$
where $\mathbf{w} \equiv \sum_j \sum_{\mathbf{y}' \in S_j} \alpha_{j\mathbf{y}'} \delta\Psi_j(\mathbf{y}')$.
6: compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in Y} H(\mathbf{y})$
7: compute $\xi_i = \max\{0, \max_{\mathbf{y} \in S_i} H(\mathbf{y})\}$
8: **if** $H(\hat{\mathbf{y}}) > \xi_i + \epsilon$ **then**
9: $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
10: $\alpha_S \leftarrow$ optimize dual over S, $S = \cup_i S_i$.
11: **end if**
12: **end for**
13: **until** no S_i has changed during iteration

Figure 2.1: General training algorithm for structured SVMs.

2.1.2 Loss Function

The MERT algorithm performs an optimization of the log-linear parameters in order to obtain the translation which maximizes the BLEU [18] score. Specifically, the BLEU score measures the precision of unigrams, bigrams, trigrams, and 4-grams between two sentences. Since the BLEU measure is a score instead of an error rate, the following loss function is used:

$$\Delta(\mathbf{e}_n, \mathbf{e}) \; = \; 1 - BLEU(\mathbf{e}_n, \mathbf{e}) \; . \tag{2.7}$$

As said in this section, the training algorithm (Figure 2.1) minimizes the training loss, so BLEU will be maximized.

There is a small problem with BLEU when applied to a pair of sentences alone, it returns a zero although the pair of sentences are very similar or even the same. For example, this pair of sentences [*"No smoking ."*,*"No smoking ."*] have a BLEU score equal to zero, because the pair does not contain any common 4-gram, although the sentences are identical. The same occurs for very similar sentences as [*"The red house is near ."*,*"The red home is near ."*]. Using the BLEU score not allow (in some cases) to distinguish between a similar sentence and a totally different one.

BLEU is used because the MERT algorithm attempts to maximize the BLEU score and so we do. This way we are able to compare the performance of both algorithms. Other measures, as for example, TER (Translation Edit Rate) [20] or WER (Word Error Rate), can be used as well.

2.1.3 Maximization

While the modeling of the feature mapping and the loss function is more or less straightforward, solving the maximization problem typically requires exploiting the structure of the output values.

In our case, the maximization is stated as follows:

$$\hat{\mathbf{e}} = \underset{\mathbf{e} \in E}{\text{argmax}} \, H(\mathbf{e}) \; . \tag{2.8}$$

Among all possible target sentences E, we have to be able to choose that one which maximizes $H(\mathbf{e})$. The set of all possible target sentences is infinite so we approximated this maximization by using n-best lists.

2.2 Implementation Details

This section describes the implementation details of the proposed optimization algorithm. In our implementation, publicly-available well-known software in the field of SMT has been used.

To calculate the feature function $\Psi(\mathbf{e}, \mathbf{f})$, the score of each model in the log-linear combination for the pair of sentences has to be computed. To calculate these scores we have used an extension of the THOT toolkit [32], which is a toolkit for SMT to train phrase-based models. The above mentioned extension of the THOT toolkit allows to obtain the alignment for a pair of sentences which maximizes the probability given by the log-linear model. It uses the current vector of weights $\boldsymbol{\omega}$ (see Section 2.1) to calculate this alignment and returns the score of each model for this pair of sentences given this alignment.

Regarding the maximization problem described in section 2.1.3, given a source sentence $\mathbf{e}_i$ we have used the MOSES toolkit [33] to calculate a n-best list of translations according to the current vector of weights $\boldsymbol{\omega}$. Then these n-best hypothesis are re-scored according to $H(\mathbf{e})$, and the one with the maximum score is returned as the required target sentence.

The THOT toolkit and the MOSES toolkit use slightly different translation tables. Specifically, the MOSES toolkit allows to work with one or more score components for each phrase pair while the THOT toolkit only allows to work with one. By this reason, it is necessary to keep two translation tables, one for the MOSES toolkit where the score for each component appears separately and one for the THOT toolkit where all the components are gathered in only one value.

2.3 Experimental Setup

We have carried out an experimentation in order to verify the effectiveness of our proposal. In our experiments we have compared the performance of both the MERT procedure and our proposed technique. All the experiments have been carried out with the FUB corpus. The FUB corpus [34], is a bilingual Italian–English corpus with a restricted semantic domain. The application is the translation of queries, requests and complaints that a tourist may make at the front desk of a hotel, for example, asking for a booked room, requesting a service of the hotel, etc. The statistics of the corpus are shown in Table 2.1.

	Training		Development		Test	
Language	Italian	English	Italian	English	Italian	English
Sentences	2900		138		300	
Run. words	51902	62335	2609	3119	6121	7243
Voc.	2480	1671	534	443	715	547
OoV	—		55	31	129	84
Perplexity	—		19.9	10.6	19.6	10.2

Table 2.1: Statistics of the FUB corpus. OoV stands for ”Out of Vocabulary” words.

The language models used in our experimentation were computed with the SRILM [35] toolkit, using 3-grams and applying interpolation with the Kneser-Ney discount. The perplexity of the corpus, according to these language models, are also shown in Table 2.1.

The evaluation has been carried out using the WER and BLEU measures, following previous works in statistical machine translation and for comparison purposes. The WER criterion is similar to the edit distance

used in Speech Recognition. It computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the sentence considered ground truth. The BLEU measure [18] computes the precision of unigrams, bigrams, trigrams and 4-grams with respect to the reference translation with a penalty for too short sentences.

2.4 Experiments

The experimentation consists on training a SMT log-linear model with the MOSES toolkit using the Training set. Then the Development set is used to optimize the parameters of the trained log-linear model. The MERT procedure and our algorithm are used to perform the optimization. Finally the translation results of each of them with the Test set are compared.

As first step, a log-linear model is trained using the MOSES toolkit. This log-linear combination is composed of eight models: the distortion (reordering) model, the target language model, the translation model which is also composed of five sub-models and the word penalty model.

As said in Section 2.2, the THOT toolkit works with translation tables with only one score for each phrase pair. So a new translation table has to be built. The score of a phrase pair in that table is the weighted average (the MOSES default weights are used) of the five scores in the MOSES translation table for that phrase pair. Once the translation scores have been gathered, a log-linear combination of four models is obtained. The new table with the gathered scores is used to perform the optimization of parameters.

To optimize the parameters the MERT procedure is used with its default options values. It uses a 100-best list of translations.

Our proposal uses the extension of the THOT toolkit to perform the feature mapping. The maximization described in Section 2.1.3 is carried out using 10-best lists of translations. The 10-best translations list is re-scored using the following equation:

$$H(\mathbf{e}) = \Delta(\mathbf{e}_n, \mathbf{e}) - \langle \delta\Psi_n(\mathbf{e}) \cdot \boldsymbol{\omega} \rangle \; . \tag{2.9}$$

This $H(\mathbf{e})$ function corresponds to the margin re-scaling ($SVM_1^{\Delta m}$) on Figure 2.1 [23].

SVM^{struct} allows to modify a great amount of parameters relative to the SVMs optimization process. Different combinations of values of some parameters have been tested to choose those values with better performance.

Table 2.2 shows the BLEU scores for the different models after translating the Test set. MERT corresponds with the model after being optimized using the MERT procedure and SVMs corresponds with a model which parameters had been optimized using our proposal.

MERT	SVMs
64.93	**65.38**

Table 2.2: BLEU translation score for the MERT algorithm and our SVM based proposal.

The results on Table 2.2 show that our proposal is able to outperform the MERT procedure. But, if we optimize the parameters using the MERT procedure and the original table (the one with eight scores per phrase pair), the BLEU score raises to 65.89. In this case, the MERT procedure is able to optimize the weights for each of the sub-models in the translation model independently. So, the relative significance of each of this sub-models can vary. Our proposal optimizes the weight of the gathered translation model, so the relative importance of each of the sub-models do not change respect to the non-optimized model.

2.5 Conclusions

This chapter has introduced a new method to optimize the parameters of a log-linear translation model using SVMs. Our proposal is based on the SVM^{struct} algorithm which is an SVM optimization algorithm for multivariate or structured outputs. The obtained results are very promising: using only a 10-best translations list, we outperform the MERT procedure when using equal number of components in the log-linear combination.

As future work, our main goal is to compare our proposal with a standard implementation of the MERT procedure in terms of time complexity; to achieve such a goal it is necessary to integrate the functionalities of the THOT, MOSES and SVM^{struct} toolkits, so the efficiency of the algorithm will be dramatically increased. In addition, we also plan to accomplish experiments with larger corpora, to use other measures as WER or TER as loss function, to use word graphs instead of n-best lists to perform the maximization and finally to find the best way to go through the differences between the THOT toolkit and the MOSES toolkit.

CHAPTER 3

A NOVEL ALIGNMENT MODEL INSPIRED ON IBM MODEL 1

In this chapter we present a novel word alignment model (Section 3.1) intended to overcome some of the problems inherent to IBM Model 1 (Section 1.1). The experimentation carried out is intended to study the behavior of our proposal in two different ways. On the one hand, we will use our model to align sentences in a paragraph aligned corpus in order to obtain a sentence aligned corpus (Section 3.2). On the other hand, we will show that an improvement in translation quality, on the Europarl corpus, can be achieved when using our proposed model as one more information source in a log-linear machine translation model (Section 3.3).

3.1 Model Description

Our alignment model is an enhancement of the IBM Model 1, which takes into account a given segmentation of the input and output sentences to estimate a statistical dictionary. The aim of our model is to benefit those alignments which are coherent with a fixed given segmentation which is considered optimal. We expect to reduce the dispersion of the lexical probabilities, concentrating the probability mass in those words which are revealed by the segmentation as potential candidates to be a correct translation. In addition, our model also aims to reduce the "garbage words" problem of IBM Model 1, which tends to concentrate alignment points in some words, independently of the distance between source and target words.

We are given a source sentence $\mathbf{X}$ divided into K segments $\mathbf{X} = X_1 \dots X_k \dots X_K$, where each segment X_k is a sequence of Γ_k words $X_k = x_{k1} \dots x_{kk'} \dots x_{k\Gamma_k}$. This source sentence is to be translated into a target sentence $\mathbf{Y}$ which is divided into L segments $\mathbf{Y} = Y_1 \dots Y_l \dots Y_L$, where each segment Y_l is a sequence of Λ_l words $Y_l = y_{l1} \dots y_{ll'} \dots y_{l\Lambda_l}$. The segmentation of the source and target sentences is given as input for our model and remains fixed throughout all the process.

In order to take into account the segmentations of the input and output sentences, we modify the statistical alignment model in Eq. (1.2) as follows:

$$Pr(\mathbf{X}|\mathbf{Y}) = \sum_{\mathbf{c},\mathbf{b}} Pr(\mathbf{c}|\mathbf{Y}) Pr(\mathbf{X},\mathbf{b}|\mathbf{Y},\mathbf{c}) \ . \tag{3.1}$$

Instead of only considering one 'hidden' word alignment $\mathbf{a}$, as IBM Model 1 does, our proposal has two 'hidden' alignments. First, we introduce a segment alignment $\mathbf{c} = c_1 \dots c_k \dots c_K$, which describes a mapping from a source segment k to a target segment $l = c_k$. Once the segment alignment is determined, we include a word alignment $\mathbf{b} = b_1 \dots b_k \dots b_K$, $\forall k \ b_k = b_{k1} \dots b_{kk'} \dots b_{k\Gamma_k}$ which describes a mapping from the k'th word of

source segment k to the l'th word of target segment l, with $l' = b_{kk'}$. Hence, alignment $\mathbf{c}$ maps a given source segment into a specific target segment, and then alignment $\mathbf{b}$ maps the words on the source segment into the words in the target segment.

3.1.1 Model Assumptions

Next, we describe the assumptions made in the derivation of our model. First, the second term on Eq. (3.1) is analyzed, on Eq. (3.2) we assume that the alignment of a given segment does not depend on the alignment of the previous segments, whereas on Eq. (3.3) we perform a similar assumption on the word level, i.e. the alignment of a given word does not depend on the previous word alignments.

$$\begin{aligned} Pr(\mathbf{X}, \mathbf{b}|\mathbf{Y}, \mathbf{c}) &= \prod_{k=1}^{K} Pr(X_k, b_k|\mathbf{Y}, \mathbf{c}, X_1^{k-1}, b_1^{k-1}) \approx \prod_{k=1}^{K} p(X_k, b_k|\mathbf{Y}, c_k) && (3.2) \\ &= \prod_{k=1}^{K} \prod_{k'=1}^{\Gamma_k} p(x_{kk'}, b_{kk'}|\mathbf{Y}, c_k, x_1^{k'-1}, b_1^{k'-1}) \\ &\approx \prod_{k=1}^{K} \prod_{k'=1}^{\Gamma_k} p(x_{kk'}, b_{kk'}|\mathbf{Y}, c_k) \; . && (3.3) \end{aligned}$$

The same assumption done on Eq. (3.2) can be applied to the first term on Eq. (3.1), yielding

$$Pr(\mathbf{c}|\mathbf{Y}) = \prod_{k=1}^{K} Pr(c_k|\mathbf{Y}, c_1^{k-1}) \approx \prod_{k=1}^{K} p(c_k|\mathbf{Y}) \; . \quad (3.4)$$

Lastly, we will perform the same assumption as IBM Model 1, modelling the mappings between input and output positions in the alignments as uniform distributions.

3.1.2 Our Model

The final formulation of our model is shown in Eq. (3.5) and Eq. (3.6):

$$p(X|Y) = \prod_{k=1}^{K} \left[\frac{1}{L+1} \sum_{l=0}^{L} p(X_k|Y_l) \right] \; . \quad (3.5)$$

$$p(X_k|Y_l) = \prod_{k'=1}^{\Gamma_k} \left[\frac{1}{\Lambda_l + 1} \sum_{l'=0}^{\Lambda_l} p(x_{kk'}|y_{ll'}) \right] \; . \quad (3.6)$$

Our model can be seen as a composition of two models: the first component (equation (3.5)) models the mapping between the segments of the input and output sentences ($\mathbf{c}$ alignment) while the second one (equation (3.6)), which is embedded into Eq. (3.5), models the alignment between the words of one source segment and the words in the corresponding target segment ($\mathbf{b}$ alignment). However, it is important to point out that both components are estimated jointly and build up our entire model.

As the standard IBM Model 1, the parameters of our model constitute a statistical word dictionary $p(x_{kk'}|y_{ll'})$.

We use the Expectation-Maximization (EM) algorithm [36] to obtain the maximum-likelihood estimates of the parameters.

The parameter re-estimation process in the EM algorithm shows the differences between our model and IBM Model 1. IBM Model 1 obtains the expected value for an alignment with the following equation [5]:

$$a_{nji}^{(t)} = \frac{p(x_{nj}|y_{ni})^{(t)}}{\sum_{i'=0}^{I} p(x_{nj}|y_{ni'})^{(t)}} \,. \tag{3.7}$$

In our case, we took into account the segmentation of the input and output sentences to obtain the expected value for an alignment, yielding the following equation:

$$(c_{nkl} \cdot b_{kll'})^{(t)} \;=\; \frac{p(x_{nkk'}|y_{nll'})^{(t)}}{\sum_{l''=0}^{\Lambda_l} p(x_{nkk'}|y_{nll''})^{(t)}} \cdot \frac{p(X_k|Y_l)}{\sum_{l'''=0}^{L} p(X_k|Y_{l'''})} \,. \tag{3.8}$$

In the original IBM Model 1 (equation (3.7)) each word alignment has the same significance, no matter the positions of the words. In our formulation (equation (3.8)) the importance of each word alignment is weighted by the significance of the alignment of the segments the words belong to with respect to the rest of segment alignments. Hence, we benefit those alignments coherent with the given segmentation which is considered optimal.

3.2 Sentences Alignment

The aim of this experimentation is to evaluate the goodness of the proposed alignment model. Our model is used to align the sentences of a paragraph aligned corpus.

3.2.1 Experimental Setup

We perform our experimentation on the METEUS [37] corpus. The METEUS corpus is a paragraph aligned corpus, which is built from 28 months of daily weather forecasts reports in the Spanish and Basque languages. These reports were picked from those published in Internet by the Basque Institute of Meteorology. To evaluate our model, a test set has been developed by hand-aligning the sentences of a set of 20 paragraphs. The segmentation and alignment of the corpus by hand requires a huge amount of work, so we use such a little test set. The statistics of the METEUS corpus are displayed in Table 3.1.

	Training		Test	
	Basque	Spanish	Basque	Spanish
Paragraphs	2178		20	
Sentences	10268	9576	82	79
Run. words	119827	135356	1015	1159
WpP	55.0	62.2	50.7	57.9
Vocabulary	1362	748	217	178

Table 3.1: Statistics for METEUS corpus. WpP stands for Words per Paragraph.

Basque is a pre-Indoeuropean language of unknown origin. Thus, the etymology of words in Basque and Spanish is usually different. It also presents a different arrangement of th words within phrases, since, unlike Spanish, Basque has left recursion. Notice that the Basque language vocabulary size for this task is 1.8 times higher than the Spanish one. This is not unusual given the Basque language is a highly inflected language, in both nouns and verbs. These great linguistic differences between these two languages make Spanish–Basque (Es–Ba) machine translation to be a difficult task for SMT.

To train our model, we previously need a segmentation of the corpus (see Section 3.1). In this task, as we want to sentence align a paragraph aligned corpus, we use the period "." as sentence separator. This naive segmentation results in the number of sentences displayed in Table 3.1.

The evaluation has been carried out using the Precision, Recall and F-Mean measures. These are automatic evaluation measures respect to a reference alignment. Given one alignment A and a reference alignment A_r, Precision, Recall and F-Mean are defined as:

$$\text{Precision} = \frac{|A \cap A_r|}{|A|} \qquad \text{Recall} = \frac{|A \cap A_r|}{|A_r|} \qquad \text{F-Mean} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.9}$$

3.2.2 Experiments

For each language direction, we trained one of our alignment models using the training set. Each model is used to align the sentences of the paragraphs of the test set. The alignments are obtained using a maximum probability approximation, so they behave as functions, i.e. one segment in the source language is aligned with the segment of the target with a maximum probability (N to 1 alignment), so there are alignments that can not be obtained (1 to N or N to N). This is a problem inherit from the IBM Model 1. Because of that, our model has an inherent loss in Recall.

To solve this problem we obtain new alignments by combining different alignments as done in [38]. Models in both directions of translation are trained and direct A_D and inverse A_I alignments are calculated. To try to obtain a better alignment, A_D and A_I are combined in a new mixed alignment A_M. We use the Intersection ($A_M = A_D \cap A_I$) and Union ($A_M = A_D \cup A_I$) operations. Intersection produces an alignment of higher Precision and lower Recall while Union results in lower Precision and higher Recall.

Due to comparison purposes we report also results with the RecAlign [39] algorithm. RecAlign is a greedy algorithm directly based on a statistical translation dictionary. RecAlign performs recursively to compute an alignment in a parallel corpus using a set of *Anchor words* as separators between segments.

Table 3.2 shows the alignment quality for test set as measured by Precision, Recall and F-Mean. We report results of our model for each direction of translation (Es–Ba and Ba–Es), for the mixed alignments (Union and Intersection) and for the RecAlign algorithm.

	Precision	Recall	F-mean
Es–Ba	0.94	0.87	0.91
Ba–Es	0.91	0.88	0.90
Union	0.88	**0,99**	**0,93**
Intersection	**0,99**	0.76	0.86
RecAlig	**1.0**	0.97	**0.98**

Table 3.2: Precision, Recall and F-Mean results for test set alignments.

Although our proposal obtains alignments that can not outperform the ones from the RecAlign algorithm, the Es–Ba and Ba–Es alignments results are no too bad. As we expect Recall values are alway lower than the Precision ones but not too much, probably because most of the alignments are N to 1 type. Mixed improvements also behave as we expect. On the one hand, Union improves the Recall measure but it decreases the Precision resulting on a slight improvement for F-Mean. On the other hand, Intersection improves precision but decreases Recall resulting on a worse F-Mean score. The better performance of the RecAlign algorithm is not a surprise given that RecAlign is a far more complex algorithm than our proposal.

3.3 Improving Translation Quality

In our experimentation we include scores derived from our model into a log-linear combination, as another feature functions, with the purpose of improving the translation quality of the log-linear model.

3.3.1 Experimental Setup

We perform this experiments on the second version of the Europarl corpus [40], which is built from the proceedings of the European Parliament. This corpus is divided into three separate sets: one for training, one for development and one for test and was the corpus used in the 2006 Workshop on Machine Translation (WMT) of the ACL [41]. We focused on the German–English (De–En), French–English (Fr–En) and Spanish–English (Es–En) subcorpora of the Europarl corpus, as done in the 2006 WMT of the ACL.

		De	En	Es	En	Fr	En
Training	Sentences	751K		731K		688K	
	Run. words	15.3M	16.1M	15.7M	15.2M	15.6M	13.8M
	Avg. len.	20.3	21.4	21.5	20.8	22.7	20.1
	Voc.	195K	66K	103K	64K	80K	62K
Development	Sentences	2000		2000		2000	
	Run. words	55K	59K	61K	59K	67K	59K
	Avg. len.	27.6	29.3	30.3	29.3	33.6	29.3
	OoV	432	125	208	127	144	138
Test	Sentences	2000		2000		2000	
	Run. words	54K	58K	60K	58K	66K	58K
	Avg. len.	27.1	29.0	30.2	29.0	33.1	29.3
	OoV	377	127	207	125	139	133

Table 3.3: Statistics of the Europarl corpus for each of the subcorpora. OoV stands for ”Out of Vocabulary” words, K for thousands of elements and M for millions of elements.

Since the original corpus is not sentence-aligned, different corpora are obtained while building the parallel bilingual corpora. The statistics of these corpora are displayed in Table 3.3. The language models used in our experimentation were computed with the SRILM [35] toolkit, using 5-grams and applying interpolation with the Kneser-Ney discount. The perplexity of the various subsets of the corpora, according to these language models, are shown in Table 3.4.

	German	English	Spanish	French
Development	148.6	89.9	89.0	66.5
Test	149.8	88.9	90.6	66.7

Table 3.4: Perplexity of the various corpora subsets with 5-grams.

It seems important to point out the fact that the average sentence length in the training sets is much shorter than in the other sets is because in the cited workshop the training sets were restricted to sentences with a maximum length of 40 words, whereas the rest of sets did not have this restriction.

Since the translations in the corpus have been written by a big number of different human translators, a same sentence may be translated in several different ways, all of them correct. This fact increases the difficulty of the corpus, and can be seen in the number of different pairs that constitute the training set, which is very similar to the total number of pairs, and also worsens the problem of ”garbage collector” words, which our model attempts

to reduce. An example is the English sentence *"We shall now proceed to vote."*: it appears translated into Spanish both as *"Se procede a la votación."*, which is quite a faithful translation, and *"El debate queda cerrado."*, which means *"the debate is now closed."*. Although these two Spanish sentences are clearly different, one can easily imagine a scenario where both translations would fit.

To train our models, we previously need a segmentation of the corpus (see Section 3.1). There are a number of algorithms to segment a corpus [42, 43, 10]. In our case, the segmentation was obtained following the technique described in [44]. First, a phrase-based model trained on a training set is used to translate the training set itself. Then, the alignment inherent to the translation of each sentence pair of the training set is used to segment this sentence pair. The resulting segmented corpora is used by our model as input.

The evaluation has been carried out using the WER and BLEU measures, following previous works in statistical machine translation and for comparison purposes. The WER criterion is similar to the edit distance used in Speech Recognition. It computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the sentence considered ground truth. The BLEU measure [18] computes the precision of unigrams, bigrams, trigrams and 4-grams with respect to the reference translation with a penalty for too short sentences.

To test the statistical significance of the results, we have followed the methods described in [45] and [46]. Zhang and Vogel present a bootstrap [47] based algorithm that computes a confidence interval, based on bootstrap percentiles, for the discrepancy between the two machine translation systems (X and Y) under study. This algorithm extracts N bootstrap samples $T_1 \dots T_i \dots T_N$ from the translated test set. If system X scored x_i on T_i and system Y scored y_i, then the discrepancy score between system X and Y on T_i is $\delta_i = x_i - y_i$. From the N discrepancy scores, we find the 2.5th percentile and the 97.5th percentile, which is the 95% confidence interval for the discrepancy between the systems. Bisani and Ney present a similar method where instead of returning and interval they compute the *Paired Probability of Improvement* (PPoI) which is the relative number of times system X outperforms system Y and vice versa.

3.3.2 Experiments

For each language pair, we trained two of our alignment models on the corresponding segmented training set, one model for each translation direction. These will be called, hereafter, our direct and inverse extended lexicalised models.

We used the Moses toolkit [33] to train the phrase-based models from the training subcorpora of Europarl and the parameters of the log-linear models were optimized using the development subcorpora via the MERT [17] procedure, using BLEU as the measure to be optimized.

The standard Moses translation model includes five translation scores for each phrase pair in the phrase table [48]: two phrase translation scores (direct and inverse), based on counting the co-occurrences of each phrase pair and normalizing the counts, two lexical weights, whose purpose is to assert the lexical soundness of each bilingual phrase pair, and a constant value called phrase penalty.

Similarly, we can obtain two lexical probabilities given by the likelihood of the phrase pair $[X_k, Y_l]$ according to our direct and inverse extended lexicalised models (equation (3.6)).

Table 3.5 shows the translation quality for the test set as measured by BLEU and WER. *Baseline* stands for the standard Moses log-linear translation model, whereas the *Extended* combination is obtained by including the direct and inverse scores of our extended lexicalised models into the *Baseline* system. Results are shown for both *monotonic* and *non monotonic* decoding. In this context, *monotonic* implies that both the segmentation of the training set and the final translation of the test set were performed monotonically. In contrast, *non monotonic* implies that both the segmentation and the translation were performed using the standard lexicalised reordering implemented into Moses.

The inclusion of our lexicalised models is reflected in an improvement of the translation quality, as measured

	Monotonic				Non Monotonic			
	Baseline		Extended		Baseline		Extended	
Language Pair	WER	BLEU	WER	BLEU	WER	BLEU	WER	BLEU
Es-En	58.25	31.01	**57.87**	**31.27**	57.67	31.56	**57.35**	**31.99**
En-Es	59.50	30.16	**59.26**	**30.52**	58.37	31.26	**58.23**	**31.54**
De-En	66.82	25.00	**66.71**	**25.01**	65.45	26.21	**65.06**	**26.49**
En-De	72.45	18.04	**71.71**	**18.42**	71.57	18.81	**71.33**	**18.92**
Fr-En	57.67	30.83	**57.59**	**30.99**	57.34	31.46	**57.08**	**31.71**
En-Fr	60.50	32.31	**60.41**	**32.37**	59.17	33.34	**58.76**	**33.75**

Table 3.5: BLEU and WER translation results for test set. Baseline stands for the standard Moses log-linear model, Extended for the standard Moses log-linear combination plus the two (direct and inverse) scores of our models, Monotonic for monotonic decoding and Non Monotonic for non monotonic decoding.

by WER and BLEU scores, both in the monotonic and the non monotonic cases. Our interpretation for this fact is that the model presented here incorporates further information into the log-linear combination of models, which is evidenced by a slight, but systematic, improvement in the translation quality over all the language pairs.

	BLEU				WER			
	Monotonic		Non Monotonic		Monotonic		Non Monotonic	
Lang. Pair	Improvement	PPoI	Improvement	PPoI	Improvement	PPoI	Improvement	PPoI
Es–En	**0.26±0.23**	**0.98**	**0.43±0.24**	**1.00**	**-0.38±0.21**	**1.00**	**-0.31±0.23**	**0.99**
En–Es	**0.36±0.26**	**0.99**	**0.28±0.23**	**0.99**	**-0.22±0.22**	**0.97**	-0.16±0.22	**0.85**
De–En	-0.03±0.18	0.35	**0.27±0.27**	**0.97**	-0.10±0.23	**0.85**	**-0.36±0.28**	**0.99**
En–De	**0.38±0.21**	**1.00**	0.09±0.25	0.79	**-0.72±0.25**	**1.00**	-0.27±0.28	**0.94**
Fr–En	**0.18±0.18**	**0.98**	**0.23±0.20**	**0.99**	-0.07±0.17	**0.82**	**-0.25±0.21**	**0.99**
En–Fr	0.05±0.23	0.73	**0.43±0.27**	**1.00**	-0.10±0.28	0.66	**-0.41±0.27**	**1.00**

Table 3.6: Average improvements with their confidence intervals at 95% and Paired Probabilities of Improvement (PPoI) of the Extended model with respect to the Baseline model, for both BLEU and WER measures. Bold improvements are statistically significant, and bold PPoIs reflect a real superiority of the Extended model.

Table 3.6 shows the average improvements with their confidence intervals, at a confidence level of 95%, of the Extended models with respect to the Baseline models for each of the language pairs considered and considering both the monotonic and non monotonic cases, following the technique described in [45]. Table 3.6 also displays the PPoI of the Extended system versus the Baseline system, according to [46].

Most of the results for non monotonic decoding show an improvement with confidence intervals that do not overlap with zero, so we can claim that the Extended model is statistically better than the Baseline model [45] for almost all the language pairs when using non monotonic decoding, and even in those cases where the improvement in the translation quality is not statistically significant the PPoI ranges between 0.8 and 1.0 so we can be confident that results reflect a real superiority of the Extended model [46]. On the other hand, when performing monotonic decoding, differences are statistically significant in less cases, and PPoI is, in general, lower than in the non monotonic case. This is due to the fact that, in our model, there is a correlation between the quality of the given segmentation of the corpus and the quality of the statistical dictionary estimated by our model. As the quality of the non monotonic segmentation is better than the quality of the monotonic one [44], our statistical dictionary is better estimated for the non monotonic case.

For both monotonic and non monotonic, translation quality results of the Extended model improve the Baseline model. However, a statistical dictionary allowing a significant improvement over the Baseline system was obtained only when the quality of the segmentation of the corpus was improved. This is specially interesting,

given that the segmentation used is defined in [44] as *approximated* segmentation, and hence further improvements cannot be discarded if the segmentation is improved as well.

3.4 Conclusions

In this chapter a novel alignment model has been introduced, which enhances IBM Model 1 by including information about a fixed given segmentation of the input and output sentences in the estimation process of the statistical dictionary. This model has been used to align into sentences a paragraph aligned corpus and, in combination with other models, to improve the translation quality as measured by BLEU and WER on the Europarl corpus. Although our proposal do not improves, as measured by Precision, Recall and F-Mean, the alignments obtained with other algorithms, results obtained, when our model is incorporated as a new feature function in a log-linear combination systematically improve baseline BLEU and WER scores. In addition most of these improvements are statistically significant or reflect a real superiority of the Extended model.

Our proposal is a first step towards a hybrid word and phrase based alignment model. Future work includes further research on the correlation of the quality of the statistical dictionary with the quality of the segmentation by trying out different segmentations. Within this line, the final aim is to calculate the statistical dictionary and simultaneously estimate the best segmentation of the corpus, instead of using a given one.

CHAPTER 4

LATIN–SPANISH TRANSLATION TASK

There are huge historical document collections residing in libraries, museums and archives that are currently being digitalized for preservation purposes and to make them available worldwide through large, on-line digital libraries. The main objective, however, is not to simply provide access to raw images of digitized documents, but to annotate them with their real informative content and, in particular, with text transcriptions and, when convenient, text translations too.

Documents in historical collections are written in old forms of current official languages and also in dead languages. It is often desirable to translate transcribed texts into current, official languages to facilitate their understanding. Unfortunately, current MT techniques are still far from being error-free, and thus they cannot produce acceptable translations in a fully automatic way. Thus, a person-machine collaborative model has to be followed so as to produce high-quality document interpretation in a cost-effective way.

To increase the productivity of the translation process, human correction activities are incorporated within the translation process itself in a *computer-assisted translation* system [49]. The idea is to use a MT system to produce portions of the target sentence that can be accepted or amended by a human translator and these correct portions are then used by the MT system as additional information to achieve further, hopefully improved suggestions.

To apply this framework we need first to build a MT system between the pair of languages to be translated. The aim of this chapter is to describe the process that leads to the creation of such a MT system. Specifically, we are going to build a MT system to translate between Latin (La) and Spanish (Es). First, we need a bilingual aligned corpus. The goal of this chapter are, on the one hand, to create a suitable bilingual corpus for a text translation task from Latin to Spanish (Section 4.1), and, on the other hand, to present the preliminary translation results using SMT techniques (Section 4.3).

4.1 Corpus creation: the NovaVulgata corpus

Acquisition of a parallel corpus for the use in a SMT system typically takes five steps:

- Obtain the raw data (e.g. by crawling from a web source)
- Extract and map parallel chunks of text (document alignment)
- Break the text into sentences (sentence splitting)
- Standardize the format of the text (tokenization)
- Map sentences in one language to sentences in the other language (sentence alignment)

In the following, we will describe in detail the acquisition of the Latin–Spanish corpus from the Internet.

In a first try, we search for classical Latin books and its correspondent translations. There is a number of web-pages that contain such texts[1] [2] [3] but texts contained in this web-pages are too short resulting on 'toy' corpora not suitable to build a competitive SMT system. We need a large text with available translations in Latin and Spanish, the Nova Vulgata can be such text. Religious texts such as the Nova Vulgata are widely available, carefully translated, and appear in both Latin and Spanish languages. The Nova Vulgata (Bibliorum Sacrorum nova vulgata editio, ISBN 88-209-2163-4), also called the Neo-Vulgata or Neo-Vulgate, is currently the typical Latin edition published by the See of Rome and approved for use in the liturgy. It is freely available at: http://www.vatican.va/archive/bible/index_sp.htm.

Taking all the books of The Nova Vulgata together, the corpus represents at least 30–40 authors in a variety of text styles, including representative samples of narrative, poetry, and correspondence. The New Testament subcorpus alone compares favorably in size to other major collections analyzed by scholars.

The website of the Vatican provides the Nova Vulgata in form of HTML files. Each HTML file contains one of the books that conforms the Nova Vulgata. The URL for each file contains relevant information for identification, such as the name of the book. Once the documents are downloaded they are parsed to extract the relevant text from noisy HTML. It is a cumbersome enterprise that requires constant refinement and adaptation. We process the HTML data with a Python program that uses pattern-matching to extract the chapters and verses numbers and the texts related to them. This Python HTML parser is a modified version of the one described in [50] to translate web-pages.

The alignment between documents is a trivial task in this case because of the fact that each HTML file contains one full book, so using the informations contained in the URL we can map from Latin books to Spanish easily. Moreover, using the number of the chapters, we can obtain a sort of paragraph aligned text. We cannot use the number of the verses to align because they vary from the Latin version of one chapter to the Spanish version.

Now we have to segment and align the sentences of the corpus. Sentence alignment is a hard problem and although we have a paragraph aligned text each paragraph contains a big number of sentences. To sentence align this corpus we have used the RecAlign [39] algorithm. RecAlign is a greedy algorithm directly based on a statistical translation dictionary. RecAlign performs recursively to compute an alignment in a parallel corpus using a set of *Anchor words* as separators between segments. We have segmented the corpus using three different groups of anchor words: period (”.”), period and semicolon (”.;”) and period, semicolon and colon (”.;:”). Table 4.1 shows the statistics of the *NovaVulgata* corpus after being segmented and aligned with each of the sets of anchor words.

	La			Es		
	”.”	”.;”	”.;:”	”.”	”.;”	”.;:”
sentences	19K	23K	28K	19K	23K	28K
avg. length	36	31	25	45	38	31
voc.		46K			34K	
run. words		730K			905K	
singletons		21K			15K	

Table 4.1: Statistics of the NovaVulgata corpus. K stands for thousands of elements.

[1] http://perso.wanadoo.es/jueangru/index.htm
[2] http://iessapstol.juntaextremadura.net/latin/minerva/index.html
[3] http://www.forumromanum.org/literature/table.html

4.2 The BibMaryland corpus

Right after the the NovaVulgata corpus was built , the University of Maryland make publicly available its The Bible Parallel Corpus Project [51][4]. This corpus, which is still under development, implements an intermediate-level annotation, delimiting book, chapter, and verse, for a growing collection of languages. All the different language versions of the Bible are consistently annotated according to the Corpus Encoding Standard subset of the TEI [52], which includes document type definitions (DTDs) for primary data, linguistic annotation, and alignment of parallel texts. The labels (id attributes) for elements make it possible to identify verses independent of context, by including the book and chapter in the label, e.g. "GEN:1:1" for Genesis, chapter 1, verse 1. The following examples show a single verse, Matthew 1:7, in some languages:

LATIN: <v id="MAT:1:7">Salomon autem genuit Roboam Roboam autem genuit Abiam Abia autem genuit Asa </v>

SPANISH: <v id="MAT:1:7">Salomón Engendró a Roboam; Roboam Engendró a Abías; Abías Engendró a Asa;</v>

ENGLISH: <v id="MAT:1:7">And Solomon begat Roboam; and Roboam begat Abia; and Abia begat Asa;</v>

FRENCH: <v id="MAT:1:7">Salomon engendra Roboam; Roboam engendra Abia; Abia engendra Asa;</v>

SWAHILI: <v id="MAT:1:7">Solomoni alimzaa Rehoboamu, Rehoboamu alimzaa Abiya, Abiya alimzaa Asa,</v>

SWEDISH: <v id="MAT:1:7">Salomo födde Roboam, Roboam födde Abia. Abia födde Asaf;</v>

VIETNAMESE: <v id="MAT:1:7">Salomôn sinh Roboam, Roboam sinh Abya, Abya sinh Asa, </v>

There are some problems with this corpus. On the one hand, there are books which have a dramatically different number of chapters, e.g. book of Judges has 21 chapters in Spanish but only one in Latin while book of Jude has one chapter in Spanish and 21 in Latin. On the other hand, same chapter can have different number of verses on each language, e.g. Psalms have 176 verses in Spanish with a total of 2346 words while in Latin it lasts only 7 verses with a total of 62 words. If the number of chapters in one book or the number of verses in one chapter differs between the two languages, we discard this data for quality reasons.

As for the corpus NovaVulgata we proceed to align the documents, which is trivial due to the annotation, then we also use the RecAlign algorithm to segment and align the corpus into sentences. In this case, we mark the limits between verses with a special symbol than together with the period (".") act as anchor words. We choose the period as separator because the NovaVulgata corpus segmented using only the period as anchor word is which obtains better results in the experiments to build a SMT system (see Section 4.3). Finally the statistics of the final sentence aligned corpus, which will be called hereafter BibMaryland, are shown in Table 4.2.

	La	Es
sentences	26K	
avg. length	17	24
voc.	39K	46K
run. words	475K	640K
singletons	18K	24K

Table 4.2: Statistics of the BibMaryland corpus. K stands for thousands of elements.

[4] http://www.umiacs.umd.edu/∼resnik/parallel/bible.html

The final BibMaryland corpus is more or less half the size of the NovaVulgata corpus, so NovaVulgata will be the corpus used to perform the experimentation. Next section will show the preliminary translation results obtained using the NovaVulgata corpus to build a SMT system.

4.3 Experiments

To perform the experiments we have divided the NovaVulgata corpus into three sets: one for training (85% of the sentences), one for development (5%) and one for test (10%). Sentences are picked randomly. Statistics of the corpus sets for each set of anchor words is shown in Table 4.3. The language models used in our experimentation were computed with the SRILM [35] toolkit, using 3-grams and applying interpolation with the Kneser-Ney discount. Perplexity of the various subcorpora, according to these language models, are also shown in Table 4.3.

		"."		".;"		".;:"	
		La	Es	La	Es	La	Es
Training	Sentences	16K		20K		24K	
	Run. words	622K	771K	620K	769M	621M	769M
	Avg. len.	36	45	31	38	25	31
	Voc.	43K	31K	43K	31K	43K	31K
Development	Sentences	992		1177		1440	
	Run. words	36K	45K	36K	44K	36K	44K
	Avg. len.	36	45	30	38	25	31
	OoV	1256	870	1188	855	1200	895
	Perplexity	298.8	120.6	307.4	118.6	313.3	121.3
Test	Sentences	1985		2356		2882	
	Run. words	71K	88K	73K	90K	72K	90K
	Avg. len.	35	44	31	38	25	31
	OoV	2349	1788	2496	1759	2409	1676
	Perplexity	283.8	112.4	284.5	116.7	293.3	116.7

Table 4.3: Statistics of the NovaVulgata corpus for each of the subcorpora. OoV stands for "Out of Vocabulary" words and K for thousands of elements.

We used the Moses toolkit [33] to train the phrase-based models. From the training subcorpora of NovaVulgata we train a Latin-to-Spanish SMT model. The parameters of the log-linear models were optimized using the development subcorpora via the MERT [17] procedure, using BLEU [18] as the measure to be optimized. Translation and parameter optimization were performed non monotonically. The evaluation has been carried out using the WER and BLEU measures.

	"."	".;"	".;:"
BLEU	**29.3**	5.1	4.6
WER	**60.1**	80.2	81.6

Table 4.4: BLEU and WER Latin to Spanish non monotonic translation results for each of the test sets.

Table 4.4 shows the translation quality for the test set of each set of anchor words as measured by BLEU and WER. Using the period as anchor word results in a corpus that outperforms the other anchor words sets tried. Our interpretation of this fact is that using only the period as anchor word, allows to segment and align the sentences of the corpus in a better way, than using the period together with other characters as the semicolon or the colon. This can be explained by the fact that in classic Latin there not exist punctuation marks. The punctuation marks of texts in Latin were added by modern editors, and while the use of the period is more or less straightforward, the use of colon and, specially, semicolon can vary along time and form one country to other.

4.4 Conclusions

In this chapter we have described the process to build a corpus suitable to create an SMT system for a specific translation task. To build any SMT system is necessary to have collected appropriate data to train the model. First, we have crawled the Internet searching for suitable text in both the task languages: Latin and Spanish. Then, we have downloaded the HTML files and parsed them in order to extract the relevant text. Next, we have segmented and aligned the sentences of the corpus using an heuristic approach. Same process has been carried out with an annotated corpora. The overall result of this work are two, not too large, Latin–Spanish corpora appropriate to build SMT systems.

This research is a first attempt to create a reliable Latin–Spanish corpora. Future work includes continuing crawling the Internet searching for suitable texts and look for collaboration with other research centers as Libraries or Universities in order to create a larger classic languages corpora appropriate to create robust SMT systems.

CHAPTER 5

CONCLUSIONS

Two complementary enhancements to the machine translation process has been presented in addition to a practical example of the process to create a new SMT system from scratch.

On Chapter 2, we present a new method to optimize the parameters of a log-linear SMT model, using SVMs to replace the slow and iterative MERT algorithm. Experimental results on an English–Italian corpus show a slighth improvement in the translation quality when our SVM based algorithm is applied, to the same log-linear combination, instead of the MERT technique. This work was published in:

> **J. González-Rubio**, D. Ortiz-Martinez and F. Casacuberta. Optimization of log-linear machine translation model parameters using SVMs. In *Proc. of the 8th Int. Workshop on Pattern Recognition in Information Systems (PRIS 2008)*, Barcelona (Spain), June 2008.

On Chapter 3, we present a new alignment model, which enhances the estimation process of the of IBM Model 1 parameters, by including information about a fixed given segmentation of the input and output sentences in the estimation process of the statistical dictionary. Our proposal has been used as alignment model to sentence align a paragraph aligned corpus, and as a new information source in a log-linear combination in order to improve the translation quality of the system. Although, as alignment model, our proposal do not improves the results of other models, it obtains similar results being a simpler model. As a information source, experimental results show a systematic improvement in the translation quality when our model is incorporated as a new feature function in a log-linear combination. A description of the model and the results as a information source in a log-linear combination were published in:

> **J. González-Rubio**, G. Sanchis-Trilles, A. Juan and F. Casacuberta. A novel alignment model inspired on IBM Model 1. In *Proc. of the European Machine Translation Conference (EAMT08)*, Hamburg (Germany), September 2008.

On Chapter 4, we describe the process to build a corpus suitable to be used to build a specific SMT system. We begin crawling the web searching for parallel texts in the languages we are interested in, and after download and parse the HTML files to extract the relevant text, segment and align this text into sentences, we obtain two sentence aligned corpora suitable to be used to build a SMT system.

Both enhancements have proved to improve the translation quality of the systems they have been tested in, in addition the improvements in the case of the alignment model are statistically significant for most of the language pairs under study. Respect to the Latin–Spanish corpora obtained it is a good beginning but a larger corpora must be collected in order to create a reliable SMT system to be used in a computer-assisted framework. In all cases, further research can be carried out, and hence further improvements cannot be discarded given the encouraging results already obtained.

Bibliography

[1] Brown, P., Cocke, J., Pietra, S.D., Pietra, V.D., Jelinek, F., Lafferty, J., Mercer, R., Roossin, R.: A statistical approach to machine translation. Computational Linguistics **16** (1990)

[2] Tomas, J., Casacuberta, F.: Monotone statistical translation using word groups. In: Proc. of the Machine Translation Summit VIII, Santiago de Compostela, Spain (2001) 357–361

[3] Marcu, D., Wong, W.: Joint probability model for statistical machine translation. In: EMNLP, Pennsylvania, USA (2002)

[4] Zens, R., Och, F., Ney, H.: Phrase-based statistical machine translation. In: Advances in artificial intelligence. Volume 2479. (2002) 18–32

[5] Brown, P., Pietra, S.D., Pietra, V.D., Mercer, R.: The mathematics of machine translation. In: Computational Linguistics. Volume 19. (1993) 263–311

[6] Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: (meta-)evaluation of machine translation. In: Proc. of the Workshop on Statistical Machine Translation. (2007)

[7] Vogel, S., Ney, H., Tillmann, C.: Hmm-based word alignment in statistical translation. In: Computational linguistics. (1996) 836–841

[8] Tillmann, C., Vogel, S., Ney, H., Zubiaga, A.: A dp based search using monotone alignments in statistical translation. In: Computational Linguistics. (1997) 289–296

[9] Niessen, S., Vogel, S., Ney, H., Tillmann, C.: A dp based search algorithm for statistical machine translation. In: Computational linguistics. (1998) 960–967

[10] Nevado, F., Casacuberta, F., Vidal, E.: Parallel corpora segmentation by using anchor words. In: Proc. of the of EACL 2003 workshop on EAMT, Budapest, Hungary (2003)

[11] Munteanu, D., Fraser, A., Marcu, D.: Improved machine translation performance via parallel sentence extraction from comparable corpora. In: Proc. of the HLT. (2004) 265–272

[12] Och, F., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., Radev, D.: A smorgasbord of features for statistical machine translation. In: HLT-NAACL. (2004) 161–168

[13] Brown, P., Pietra, S.D., Pietra, V.D., Goldsmith, M., Hajic, J., Mercer, R., Mohanty, S.: But dictionaries are data too. In: Proc. of the HLT. (1993) 202–205

[14] Och, F., Tillmann, C., Ney, H.: Improved alignment models for statistical machine translation. In: Proc. of the EMNLP-VLC, University of Maryland (1999) 20–2

[15] Moore, R.: Improving ibm word-alignment model 1. In: Proc. of the ACL. (2004) 518

[16] Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proce. of the ACL. (2001) 295–302

[17] Och, F.: Minimum error rate training for statistical machine translation. In: Proc. of the ACL, Sapporo, Japan (2003)

[18] Papineni, K., Kishore, A., Roukos, S., Ward, T., Zhu, W.: Bleu: A method for automatic evaluation of machine translation. In: Technical Report RC22176 (W0109-022). (2001)

[19] Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. (2005)

[20] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proc. of AMTA. (2006)

[21] Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3) (1995) 273–297

[22] Vapnik, V.: The nature of statistical learning theory. Springer-Verlag New York, Inc. (1995)

[23] Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML04, year = 2004,

[24] Weston, J., Watkins, C.: Multi-class support vector machines (1998)

[25] Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2** (2002)

[26] Joachims, T.: Training linear svms in linear time. In: KDD06. (2006)

[27] Joachims, T.: Learning to align sequences: A maximum-margin approach (2003)

[28] Joachims, T.: Making large-scale support vector machine learning practical. In: Advances in Kernel Methods: Support Vector Machines. (1998)

[29] Yu, C.J., Joachims, T., Elber, R., Pillardy, J.: Support vector training of protein alignment models. In: RECOMB. (2007)

[30] Vapnik, V.: Statistical Learning Theory. (1998)

[31] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research (2005)

[32] Ortiz-Martínez, D., García-Varea, I., Casacuberta, F.: Thot: a toolkit to train phrase-based statistical translation models. In: Tenth Machine Translation Summit. (2005)

[33] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantine, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. Proc. of the ACL (2007)

[34] Casacuberta, F., Ney, H., Och, F.J., Vidal, E., Vilar, J.M., Barrachina, S., Garcia-Varea, I., D. Llorens, C.M., Molau, S., Nevado, F., Pastor, M., Pico, D., Sanchis., A.: Some approaches to statistical and finite-state speech-to-speech translation. Computer Speech and Language (2004)

[35] Stolcke, A.: Srilm - an extensible language modeling toolkit. In: Proc. of the International Conference on Spoken Language Processing. Volume 2. (2002) 901–904

[36] Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. **39**(1) (1977) 1–38

[37] Pérez, A., Torres, I., Casacuberta, F., Guijarrubia, V.: A spanish-basque weather forecast corpus for probabilistic speech translation. In: FinTAL 2006. Volume 4139. Springer (2006) 716–725

[38] Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: ACL02. (2002)

[39] Nevado, F., Casacuberta, F.: Bilingual corpora segmentation using bilingual recursive alignments. In: Actas de las III Jornadas en Tecnologías del Habla, Valencia (November 2004)

[40] Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT Summit. (2005)

[41] Koehn, P., Monz, C.: Manual and automatic evaluation of machine translation between european languages. In: Proc. of the NAACL, New York City (2006) 102–121

[42] Michel, S., Plamondon, P.: Bilingual sentence alignment: Balancing robustness and accuracy (1996)

[43] Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics **23**(3) (1997) 377–403

[44] Sanchis-Trilles, G., Casacuberta, F.: Increasing translation speed in phrase-based models via suboptimal segmentation. In: Proc. of PRIS, Barcelona (Spain) (2008)

[45] Zhang, Y., Vogel, S.: Measuring confidence intervals for the machine translation evaluation metrics. In: Proc. of the TMI. (2004)

[46] Bisani, M., Ney, H.: Bootstrap estimates for confidence intervals in asr performance evaluation. In: IEEE Conference on Acoustics, Speech, and Signal Processing. Volume 1. (2004)

[47] Efron, B., Tibshirani, R.: An Introduction to Bootstrap. Chapman and Hall, New York (1993)

[48] Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: Proc. of the NAACL-HLT. Volume 1. (2003) 48–54

[49] Foster, G., Lapalme, G.: Unit completion for a computer-aided translation typing system. In: Machine Translation. (2000) 1–0

[50] González-Rubio, J., González, J., Lagarda, A.L., Giménez, A., Navarro, J.R., Casacuberta, F.: Translation applications under the sishitra framework. In: 3rd Language and Technology Conference. (2007)

[51] Resnik, P., Olsen, M., Diab, M.: The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. In: Computers and the Humanities, 33. (1999) 129–153

[52] Ide, N.: Corpus encoding standard: Document ces 1, version 1.4. (October. http://www.cs.vassar.edu/CES/ 1996) 129–153

APPENDIX A

TECHNICAL DESCRIPTION OF THE ALIGNMENT MODEL

A.1 Data representation

- $\mathbf{X}$, sequence of segments of text, source language.
  - $\mathbf{X} = X_1 \dots X_k \dots X_K$
  - $\forall k:\ X_k$ sequence of Γ_k words.
- $\mathbf{Y}$, sequence of segments of text, target language.
  - $\mathbf{Y} = Y_1 \dots Y_l \dots Y_L$
  - $\forall l:\ Y_l$ sequence of Λ_l words.
- $\mathbf{c}$, alignment between the segments of $\mathbf{X}$ and the ones of $\mathbf{Y}$.
  - $\mathbf{c} = c_1 \dots c_k \dots c_K$
  - $\forall k:\ c_k = c_{k0} \dots c_{kk'} \dots c_{kL}$
    $\forall k, l:\ c_{kl} \in \{0, 1\}$
  - c_{kl} expresses that input segment X_k is (or is not) aligned with the output segment Y_l.
    $c_{k0} = 1$ expresses that segment X_k is aligned to the *NULL* segment.
  - Restrictions: $\forall k:\ \sum_l a_{kl} = 1$
  - $\mathcal{C}$, set of all possible $\mathbf{c}$ alignments.
- $\mathbf{b}$, represents the alignment between the words of two segments.
  - $\mathbf{b} = b_1 \dots b_k \dots b_K$
  - $\forall k:\ b_k = b_{k1} \dots b_{kk'} \dots b_{k\Gamma_k}$
  - $\forall k, k':\ b_{kk'} = b_{kk'0} \dots b_{kk'l'} \dots, b_{kk'\Lambda_l}$ being $c_{kl} = 1$
    $\forall k, k', l':\ b_{kk'l'} \in \{0, 1\}$
  - $b_{kk'l'}$ given $c_{kl} = 1$, expresses that word $x_{kk'}$ is (or is not) aligned with word $y_{ll'}$
    $b_{kk'0} = 1$ expresses that word $x_{kk'}$ is aligned to the *NULL* word.
  - Restrictions: $\forall k, k':\ \sum_{l'} b_{kk'l'} = 1$
  - $\mathcal{B}$, set of all possible $\mathbf{b}$ alingments.

A.2 The model

A.2.1 Problem description

We want to calculate the maximum probability alignment between the segments of two sentences.

$$\hat{\mathbf{c}} = \underset{\mathbf{c}\in\mathcal{C}}{\operatorname{argmax}}\, Pr(\mathbf{c}|\mathbf{X},\mathbf{Y}) \;=\; \underset{\mathbf{c}\in\mathcal{C}}{\operatorname{argmax}}\, \frac{Pr(\mathbf{c},\mathbf{X}|\mathbf{Y})}{Pr(\mathbf{X}|\mathbf{Y})} \;=\; \underset{\mathbf{c}\in\mathcal{C}}{\operatorname{argmax}}\, Pr(\mathbf{c},\mathbf{X}|\mathbf{Y})\;. \tag{A-1}$$

To calculate $Pr(\mathbf{c},\mathbf{X}|\mathbf{Y})$ we use a random variable $\mathbf{b}$:

$$Pr(\mathbf{c},\mathbf{X}|\mathbf{Y}) \;=\; \sum_{\mathbf{b}\in\mathcal{B}} Pr(\mathbf{c},\mathbf{b},\mathbf{X}|\mathbf{Y})\;. \tag{A-2}$$

The complete model is:

$$Pr(\mathbf{c},\mathbf{b},\mathbf{X}|\mathbf{Y}) \;=\; Pr(\mathbf{b}|\mathbf{Y})\cdot Pr(\mathbf{b},\mathbf{X}|\mathbf{Y},\mathbf{c})\;. \tag{A-3}$$

Each multiplier is calculated ($\approx$ expresses that a model assumption has been taken):

$$\begin{aligned}
Pr(\mathbf{c}|\mathbf{Y}) &= \prod_{k=1}^{K} Pr(c_k|\mathbf{Y},c_1^{k-1}) && \text{(A-4)}\\
&\approx \prod_{k=1}^{K} p(c_k|\mathbf{Y}) && \text{(A-5)}\\
&= \prod_{k=1}^{K}\prod_{l=0}^{L} p(c_{kl}=1|\mathbf{Y})^{c_{kl}} && \text{(A-6)}\\
&\approx \frac{1}{(I+1)^J}\;. && \text{(A-7)}
\end{aligned}$$

$$\begin{aligned}
Pr(\mathbf{X},\mathbf{b}|\mathbf{Y},\mathbf{c}) &= \prod_{k=1}^{K} Pr(X_k,b_k|\mathbf{Y},c_k,X_1^{k-1},c_1^{k-1}) && \text{(A-8)}\\
&\approx \prod_{k=1}^{K} p(X_k,b_k|\mathbf{Y},c_k) && \text{(A-9)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k} p(x_{kk'},b_{kk'}|\mathbf{Y},c_k,x_{k1}^{k'-1},b_{k1}^{k'-1}) && \text{(A-10)}\\
&\approx \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k} p(x_{kk'},b_{kk'}|\mathbf{Y},c_k) && \text{(A-11)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L} P(x_{kk'},b_{kk'}|\mathbf{Y},c_{kl}=1)^{c_{kl}} && \text{(A-12)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L}\prod_{l'=0}^{\Lambda_l} p(x_{kk'},b_{kk'l'}=1|\mathbf{Y},c_{kl}=1)^{c_{kl}b_{kk'l'}} && \text{(A-13)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L}\prod_{l'=0}^{\Lambda_l} [\, p(b_{kk'l'}=1|\mathbf{Y},c_{kl}=1)\cdot p(x_{kk'}|\mathbf{Y},c_{kl}=1,b_{kk'l'}=1)\,]^{c_{kl}b_{kk'l'}} && \text{(A-14)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L}\prod_{l'=0}^{\Lambda_l} \left[\, \frac{1}{\Lambda_l+1}\cdot p(x_{kk'}|\mathbf{Y},c_{kl}=1,b_{kk'l'}=1)\right]^{c_{kl}b_{kk'l'}} && \text{(A-15)}\\
&= \prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L}\prod_{l'=0}^{\Lambda_l} \left[\, \frac{1}{\Lambda_l+1}\cdot p(x_{kk'}|y_{ll'})\right]^{c_{kl}b_{kk'l'}}\;. && \text{(A-16)}
\end{aligned}$$

Finally, our model remais:

Complete model:

$$p(\mathbf{X},\mathbf{c},\mathbf{b}|\mathbf{Y}) \;=\; \frac{1}{(L+1)^K}\prod_{k=1}^{K}\prod_{k'=1}^{\Gamma_k}\prod_{l=0}^{L}\prod_{l'=0}^{\Lambda_l}\left[\;\frac{1}{\Lambda_l+1}\cdot p(x_{kk'}|y_{ll'})\;\right]^{c_{kl}b_{kk'l'}} . \tag{A-17}$$

Incomplete model:

$$p(\mathbf{X}|\mathbf{Y}) \;=\; \frac{1}{(L+1)^K}\prod_{k=1}^{K}\sum_{l=0}^{L}\prod_{k'=1}^{\Gamma_k}\sum_{l'=0}^{\Lambda_l}\left[\;\frac{1}{\Lambda_l+1}\cdot p(x_{kk'}|y_{ll'})\;\right] . \tag{A-18}$$

A.3 EM

A.3.1 E step

$$\mathcal{Q}(\Theta|\Theta^{(t)}) = E_{\mathbf{c},\mathbf{b}}\Big(\sum_n \log(p(\mathbf{X}_n,\mathbf{c}_n,\mathbf{b}_n|\mathbf{Y}_n)) \mid \mathbf{X}_n,\mathbf{Y}_n,\Theta^{(t)}\Big) \tag{A-19}$$

$$= \sum_n \sum_{k=1}^{K_n}\sum_{k'=1}^{\Gamma_{nk}}\sum_{l=0}^{L_n}\sum_{l'=0}^{\Lambda_{nl}}\big((c_{nkl}\cdot b_{nkk'l'})^{(t)}\log(p(x_{nkk'}|y_{nll'}))\big) + \text{const}(K,L,k,l,n)\,. \tag{A-20}$$

The expected value of $\mathbf{c}$ and $\mathbf{b}$ is:

$$(c_{nkl}\cdot b_{nkk'l'})^{(t)} = \frac{p(c_{nkl}=1,b_{nkk'l'}=1,\mathbf{X}_n|\mathbf{Y}_n)}{p(\mathbf{X}_n|\mathbf{Y}_n)}\,. \tag{A-21}$$

The numerator is derivated:

$$p(c_{nkl}=1,b_{nkk'l'}=1,\mathbf{X}_n|\mathbf{Y}_n) = p(c_{nkl}=1|\mathbf{Y}_n)\cdot p(b_{nkk'l'}=1,\mathbf{X}_n|\mathbf{Y}_n,c_{nkl}=1) \tag{A-22}$$

$$\approx \frac{1}{L_n+1}\cdot p(b_{nkk'l'}=1,\mathbf{X}_n|\mathbf{Y}_n,c_{nkl}=1) \tag{A-23}$$

$$= \frac{1}{L_n+1}\cdot p(b_{nkk'l'}=1|\mathbf{Y}_n,c_{nkl}=1)\cdot(\mathbf{X}_n|\mathbf{Y}_n,a_{nkl}=1,b_{nkk'l'}=1) \tag{A-24}$$

$$\approx \frac{1}{L_n+1}\cdot\frac{1}{\Lambda_{nl}+1}\cdot p(\mathbf{X}_n|\mathbf{Y}_n,c_{nkl}=1,b_{nkk'l'}=1) \tag{A-25}$$

$$= \frac{1}{L_n+1}\cdot\frac{1}{\Lambda_{nl}+1}\cdot p(X_{nk}|\mathbf{Y}_n,c_{nkl}=1,b_{nkk'l'}=1) \cdot p(X_{n1}^{\;k-1},X_{nk+1}^{\;K_n}|\mathbf{Y}_n,c_{nkl}=1,b_{nk'l'}=1,X_{nk}) \tag{A-26}$$

$$\approx \frac{1}{L_n+1}\cdot\frac{1}{\Lambda_{nl}+1}\cdot p(X_{nk}|\mathbf{Y}_n,c_{nkl}=1,b_{nkk'l'}=1) \cdot p(X_{n1}^{\;k-1},X_{nk+1}^{\;K_n}|\mathbf{Y}_n) \tag{A-27}$$

$$= \frac{1}{L_n+1}\cdot\frac{1}{\Lambda_{nl}+1}\cdot p(x_{nkk'}|\mathbf{Y}_n,c_{nkl}=1,b_{nkk'l'}=1) \cdot p(x_{nk1}^{\;\;k'-1},x_{nkk'+1}^{\;\;\Gamma_{nk}}|\mathbf{Y}_n,c_{nkl}=1,b_{nkk'l'}=1) \cdot p(X_{n1}^{\;k-1},X_{nk+1}^{\;K_n}|\mathbf{Y}_n) \tag{A-28}$$

$$\approx \frac{1}{L_n+1}\cdot\frac{1}{\Lambda_{nl}+1}\cdot p(x_{nkk'}|y_{nll'}) \cdot p(x_{nk1}^{\;\;k'-1},x_{nkk'+1}^{\;\;\Gamma_{nk}}|\mathbf{Y}_n,c_{nkl}=1) \cdot p(X_{n1}^{\;k-1},X_{nk+1}^{\;K_n}|\mathbf{Y}_n)\,, \tag{A-29}$$

Fourth multiplier on Eq. (A-29) is further derivated:

$$
\begin{aligned}
p(x_{nk1}^{\;\;k'-1}, x_{nkk'+1}^{\;\;\Gamma_{nk}} | \mathbf{Y}_n, c_{nkl}=1) &\approx \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-30)}\\
&= \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{b_{nkk''}} p(x_{nkk''}, b_{nkk''}|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-31)}\\
&= \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{b_{nkk''}} \prod_{l''=0}^{\Lambda_{nl}} p(x_{nkk''}, b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1)^{b_{nkk''l''}} && \text{(A-32)}\\
&= \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} p(x_{nkk''}, b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-33)}\\
&= \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} p(b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1) \\
&\qquad \cdot p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1, b_{nkk''l''}=1) && \text{(A-34)}\\
&\approx \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1, b_{nkk''l''}=1) && \text{(A-35)}\\
&= \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''})\,, && \text{(A-36)}
\end{aligned}
$$

Finally the numerator yields:

$$
\begin{aligned}
p(c_{nkl}=1, b_{nkk'l'}=1, \mathbf{X}_n|\mathbf{Y}_n) &= \frac{1}{L_n+1} \cdot \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk'}|y_{nll'}) \cdot p(X_{n1}^{\;\;k-1}, X_{nk+1}^{\;\;K_n}|\mathbf{Y}_n) \\
&\qquad \cdot \prod_{\substack{k''=1 \\ k''\neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''})\,. && \text{(A-37)}
\end{aligned}
$$

The denominator is derived:

$$
\begin{aligned}
p(\mathbf{X}_n|\mathbf{Y}_n) &= \sum_{c_{nk}} p(\mathbf{X}_n, c_{nk}|\mathbf{Y}_n) && \text{(A-38)}\\
&= \sum_{l=0}^{L_n} p(\mathbf{X}_n, c_{nkl}=1|\mathbf{Y}_n) && \text{(A-39)}\\
&= \sum_{l=0}^{L_n} p(c_{nkl}=1|\mathbf{Y}_n)\cdot p(\mathbf{X}_n|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-40)}\\
&= \sum_{l=0}^{L_n} \frac{1}{L_n+1}\cdot p(\mathbf{X}_n|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-41)}\\
&= \sum_{l=0}^{L_n} \frac{1}{L_n+1}\cdot p(X_{nk}|\mathbf{Y}_n, c_{nkl}=1)\cdot p(X_{n1}^{\;k-1}, X_{nk+1}^{K_n}|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-42)}\\
&\approx \frac{1}{L_n+1}\cdot p(X_{n1}^{\;k-1}, X_{nk+1}^{K_n}|\mathbf{Y}_n)\cdot \sum_{l=0}^{L_n} p(X_{nk}|\mathbf{Y}_n, c_{nkl}=1)\;, && \text{(A-43)}
\end{aligned}
$$

Third multiplier on Eq. (A-43) is further derived:

$$
\begin{aligned}
p(X_{nk}|\mathbf{Y}_n, c_{nkl}=1) &\approx \prod_{k''=1}^{\Gamma_{nk}} p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-44)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{b_{nkk''}} p(x_{nkk''}, b_{nkk''}|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-45)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{b_{nkk''}} \prod_{l''=0}^{\Lambda_{nl}} p(x_{nkk''}, b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1)^{b_{nkk''l''}} && \text{(A-46)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} p(x_{nkk''}, b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1) && \text{(A-47)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} p(b_{nkk''l''}=1|\mathbf{Y}_n, c_{nkl}=1)\cdot p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1, b_{nkk''l''}=1) && \text{(A-48)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1}\cdot p(x_{nkk''}|\mathbf{Y}_n, c_{nkl}=1, b_{nkk''l''}=1) && \text{(A-49)}\\
&= \prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1}\cdot p(x_{nkk''}|y_{nll''})\;, && \text{(A-50)}
\end{aligned}
$$

Finally the denominator yields:

$$
p(\mathbf{X}_n|\mathbf{Y}_n) = \frac{1}{L_n+1}\cdot p(X_{n1}^{\;k-1}, X_{nk+1}^{K_n}|\mathbf{Y}_n)\cdot \sum_{l=0}^{L_n}\prod_{k''=1}^{\Gamma_{nk}}\sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1}\cdot p(x_{nkk''}|y_{nll''})\;. \qquad \text{(A-51)}
$$

Equations (A-37) and (A-51) are used in Eq. (A-21). Then, after simplifying:

$$(c_{nkl} \cdot b_{nkk'l'})^{(t)} = \frac{\frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk'}|y_{nll'}) \prod_{\substack{k''=1 \\ k'' \neq k'}}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''})}{\sum_{\tilde{l}=0}^{L_n} \prod_{\tilde{k}=1}^{\Gamma_{nk}} \sum_{\tilde{\tilde{l}}=0}^{\Lambda_{n\tilde{l}}} \frac{1}{\Lambda_{n\tilde{l}}+1} \cdot p(x_{nk\tilde{k}}|y_{n\tilde{l}\tilde{\tilde{l}}})} \tag{A-52}$$

$$= \frac{\frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk'}|y_{nll'})}{\frac{1}{\Lambda_{nl}+1} \sum_{l''=0}^{\Lambda_{nl}} p(x_{nkk'}|y_{nll''})} \cdot \frac{\prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''})}{\sum_{\tilde{l}=0}^{L_n} \prod_{\tilde{k}=1}^{\Gamma_{nk}} \sum_{\tilde{\tilde{l}}=0}^{\Lambda_{n\tilde{l}}} \frac{1}{\Lambda_{n\tilde{l}}+1} \cdot p(x_{nk\tilde{k}}|y_{n\tilde{l}\tilde{\tilde{l}}})} \tag{A-53}$$

$$= \frac{p(x_{nkk'}|y_{nll'})}{\sum_{l''=0}^{\Lambda_{nl}} p(x_{nkk'}|y_{nll''})} \cdot \frac{\prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''})}{\sum_{\tilde{l}=0}^{L_n} \prod_{\tilde{k}=1}^{\Gamma_{nk}} \sum_{\tilde{\tilde{l}}=0}^{\Lambda_{n\tilde{l}}} \frac{1}{\Lambda_{n\tilde{l}}+1} \cdot p(x_{nk\tilde{k}}|y_{n\tilde{l}\tilde{\tilde{l}}})} , \tag{A-54}$$

Finally, the expected value yields:

$$(c_{nkl} \cdot b_{nkk'l'})^{(t)} = \frac{p(x_{nkk'}|y_{nll'})^{(t)}}{\sum_{l''=0}^{\Lambda_{nl}} p(x_{nkk'}|y_{nll''})^{(t)}} \cdot \frac{p(X_{nk}|Y_{nl})}{\sum_{\tilde{l}=0}^{L_n} p(X_{nk}|Y_{n\tilde{l}})} , \tag{A-55}$$

Given that $p(X_{nk}|Y_{nl})$ is defined as:

$$p(X_{nk}|Y_{nl}) = \prod_{k''=1}^{\Gamma_{nk}} \sum_{l''=0}^{\Lambda_{nl}} \frac{1}{\Lambda_{nl}+1} \cdot p(x_{nkk''}|y_{nll''}) . \tag{A-56}$$

A.4 M Step

We want to derivate the equations to re-estimate the parameters of our model.

$$\Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{Q}(\Theta|\Theta^{(t)}) \text{ assuming } \forall w : \sum_v p(v|w) = 1 \,. \tag{A-57}$$

Function $\mathcal{L}$ is defined:

$$\mathcal{L}(\Theta, \boldsymbol{\lambda}) \;=\; \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} (\; (c_{nkl} \cdot b_{nkk'l'})^{(t)} \log(p(x_{nkk'}|y_{nll'}))\;) - (\sum_w \lambda_w \sum_v p(v|w) - 1) \,. \tag{A-58}$$

Gradients and the maximum of $\mathcal{L}$ function are calculated.

$$\frac{\partial \mathcal{L}(\Theta, \boldsymbol{\lambda})}{\partial \lambda_w} = \sum_v p(v|w) - 1 \qquad \Longrightarrow \qquad \sum_v p(v|w) = 1 \,. \tag{A-59}$$

$$\frac{\partial \mathcal{L}(\Theta, \boldsymbol{\lambda})}{\partial p(v|w)} \;=\; \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} \frac{(c_{nkl} \cdot b_{nkk'l'})^{(t)}}{p(v|w)} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w) - \lambda_w \tag{A-60}$$

$$\Longrightarrow \;\; \lambda_w \;=\; \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} \frac{(c_{nkl} \cdot b_{nkk'l'})^{(t)}}{p(v|w)} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w) \,. \tag{A-61}$$

Using Eq. (A-61) we obtain:

$$p(v|w) \;=\; \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} \frac{(c_{nkl} \cdot b_{nkk'l'})^{(t)}}{\lambda_w} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w) \,. \tag{A-62}$$

Using Eq. (A-62) on Eq. (A-59) yields:

$$1 \;=\; \sum_v \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} \frac{(c_{nkl} \cdot b_{nkk'l'})^{(t)}}{\hat{\lambda}_w} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w) \tag{A-63}$$

$$\Downarrow$$

$$\hat{\lambda}_w \;=\; \sum_v \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} (c_{nkl} \cdot b_{nkk'l'})^{(t)} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w) \,. \tag{A-64}$$

Finally using Eq. (A-61) and Eq. (A-64), the final parameter re-estimation equation is reached, yielding

$$p(v|w)^{(t+1)} \;=\; \frac{\sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} (c_{nkl} \cdot b_{nkk'l'})^{(t)} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w)}{\sum_v \sum_n \sum_{k=1}^{K_n} \sum_{k'=1}^{\Gamma_{nk}} \sum_{l=0}^{L_n} \sum_{l'=0}^{\Lambda_{nl}} (c_{nkl} \cdot b_{nkk'l'})^{(t)} \cdot \delta(x_{nkk'}, v) \cdot \delta(y_{nll'}, w)} \,. \tag{A-65}$$

APPENDIX B

LATIN–SPANISH TRANSLATION TASK: RESOURCES LIST

To train SMT models we need parallel corpora between the languages we are translating between. All the resources listed are freely available on the Internet.

- HTTP://WWW.VATICAN.VA/ARCHIVE/BIBLE/INDEX_SP.HTM

  Official website of The Vatican, it contains the official Bible in four languages including Latin and Spanish. From this webpage was builded the NovaVulgata corpus.

- HTTP://WWW.UMIACS.UMD.EDU/ RESNIK/PARALLEL/BIBLE.HTML

  Webpage of the University of Maryland, it stores a semi-annotated corpus that is not aligned into sentences. Corpora consists on different translations of The Bible in different languages including Latin and Spanish. From this website was builded the BibMaryland corpus.
  Contact: RESNIK@UMIACS.UMD.EDU

- HTTP://CLASICAS.USAL.ES/RECURSOS/TEXTOSAUT.HTM

  Website of the Departamento de Filología Clásica e Indoeuropeo of the Universidad de Salamanca. It contains a huge collection of texts of different authors, most in their original languages (Latin and Greek). Some of them has also its translations, most of the translations are in English, few of it are also in Spanish. It is a useful website to generate language models, not so useful to translate due to the lack of Spanish translations.
  Contact: HTTP://CLASICAS.USAL.ES/PERSONALES/FCORTES.HTM
  SANA@USAL.ES.

- HTTP://IESSAPOSTOL.JUNTAEXTREMADURA.NET/LATIN/MINERVA/INDEX.HTML

  Texts and translations of this website are taken from:

  > SÁNCHEZ SALOR, E. Y CHAPARRO GÓMEZ, C. (eds.), *Francisco Sánchez de las Brozas. Minerva o de causis linguae latinae*, Cáceres, Institución Cultural El Brocense, 1995.

  This webpage contains a quite big quantity of original Latin texts and its translations to Spanish, but they are difficult to download and process. The website is organized by books and chapters.
  Contact: CARLOS.CABANILLAS@EDU.JUNTAEXTREMADURA.NET.

- HTTP://WWW.FORUMROMANUM.ORG/LITERATURE/TABLE.HTML

  Website maintained by David Camden, of the Hardvard University. It contains a big amount of original texts and quite a lot English translations, only few Spanish translations.
  Contact: CAMDEN@FAS.HARDVARD.EDU

- HTTP://PERSO.WANADOO.ES/JUAGRU/INDEX.HTM

  The webpage contains a great number of texts, generally short texts, of different classic roman authors.
  Contact: JUAGRU11@GMAIL.COM.

- HTTP://WWW.THELATINLIBRARY.COM/CLASSICS.HTML

  This website contains a lot of original classic author texts, no translations are available.
  Contact: LATINLIBRARY@MAC.COM

Another interesting resource are online translators. Following are some of the online translators available, all of them translate monotonically, word by word, using a dictionary.

- HTTP://DICTIONARIES.TRAVLANG.COM/LATINSPANISH/
- HTTP://RECURSOS.CNICE.MEC.ES/LATINGRIEGO/PALLADIUM/5_APS/ESPLAP03.HTM