Statistical Machine Translation Incorporating Morphological Knowledge

Klasinas Ioannis
Electronic and Computer Engineering
Technical University of Crete, Chania

April 9, 2008

Abstract

Machine translation between natural languages is a very challenging problem, whose efficient dealing with is very important for the free flow of information among people. Traditionally, statistical machine translation relies on the extraction of information from parallel corpora, relying solely on lexical level correspondence. As a result, linguistic information is not directly utilized.

This thesis aims at exploring an alternative way of improving the performance of a statistical translation system. Instead of using more data, which is generally not available in abundance, the use of morphological information is proposed, in order to improve the translation quality.

Different ways of incorporating morphological knowledge are tried, using a phrase-based Greek to English system as baseline. In addition to that, different ways of combining the baseline system with the morphological incorporating one are tried. The results show a small improvement in performance, up to 3% and a great reduction in the out of vocabulary words, more than 60%. All the tools and resources used for the experiments are freely available for research purposes, and are widely used by the scientific community.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine translation is by no means a new concept. The first scientific formulation of the problem can be traced bach to the first years following the end of the second world war, by Warren Weaver. Alan Turing also refers to this topic, but does not really deal with the problem. And it was for good reason that these early approaches did not materialize into experiments; translating from one natural language into another one is a task far more difficult from trying to decipher a code, or restoring a signal. It involves dealing with structural differences between languages, as in grammar and syntax, but also with conveying the meaning through the translation. While the first part may seem somewhat more easy, since grammar and syntax can be described in sufficient detail, the latter one still awaits for a satisfactory solution. The reason this problem is so difficult is that there is no clear and well defined representation of semantics, acceptable by all the speakers of a language.

The progress of this scientific field is by no means linear; there have been times of excitement, followed by long inactivity periods. This can be

attributed to the combination of two facts:

1. Dealing with a difficult problem without having enough computational resources

2. Setting the expectations too high, consequently leading to disappointment and resentment to continue such a line of research, like in [3].

Different approaches exist to machine translation. *Transfer based* uses specific language dependent rules that model the vocabulary of the language pair and the grammatical/syntactical transformations that have to be done in order to translate from one language into another. *Interlingual* brakes the translation into a two step process; the source language is first translated into a representation (interlingua) and then into the target language. In *example based* the system learns to translate from parallel corpora in various languages. In its simplest form, the sentence to be translated is compared to a set of sentences whose translation is known and one of these translations is selected for the sentence in hand.

While the above schemes have been in practise for a long time, Statistical Machine Translation (**SMT**) is a newer field, which started to become a seriously considered approach in 1990, after the seminal publication [5]. In the general case, a parallel corpus[1] is used, from which information is extracted in an unsupervised manner, a process called training. To translate a text, we search the most probable translation of the sentence at hand, with respect to the statistics gathered in the training step.

[1] A corpus translated in the two languages by hand.

SMT has proven to be a very competitive approach. The quality of the translation achieved is considered to be superior to that of the other techniques, as has been shown by a number of evaluation campaigns. It does not ask for specific linguistic knowledge, since information is extracted in an unsupervised manner from the training text. It is easy to adapt a system to a new domain or language pair, since there is no need to rewrite rules, just to use the corresponding corpus. Some limitations however do exist, and they arise from the abstract level at which an SMT system works. Traditionally, SMT systems operate strictly at the lexical level, discarding linguistic information like morphology, syntax or semantic. This means that the information available in the parallel text is only partially utilized, the rest is simply discarded, or in the best case it is only indirectly modelled. Given that bilingual corpora are hard to gather, it becomes important to be able to exploit the ones available as much as possible.

This thesis focuses on the incorporation of morphological knowledge into an SMT system. Experimentation is done using a phrase based system, dealing with the Greek-English language pair. The rest of this text is organised as follows. Chapter 2 outlines the aims of this work. Chapter 3 describes the baseline system used. Chapter 4 investigates the corpus used, and various statistics are gathered. Chapter 5 discusses the morphological features extraction. Chapter 6 describes experiments conveyed. Chapter 7 presents the results of the experiments. Chapter 8 concludes.

Chapter 2

Scientific goals

The objective of this thesis is to investigate ways of efficiently integrating morphological information into a statistical machine translation system. It is expected that this way the available resources will be better exploited and the overall performance of the translation system will improve. A constraint posed on the design of the system is that the modifications will be modular and easy to modify for use in different language pairs without requiring special tools.

Data sparseness

The fact that the system operates only at the lexical level means that different word forms are treated completely independently, even if they are in fact closely related, like different forms of a verb. This can become a severe problem if the language pair includes a morphology rich language, as is the case with Greek. In that case the ratio of distinct word forms to corpus size is quite high, meaning that many words will not be observed in the training corpus enough time to learn sufficient translation rules. The standard method

to overcome this problem is to use more training data. Gathering bilingual corpora is, however, a demanding task both in terms of time and resources needed. The situation is even worse when dealing with scarce resource languages, where even monolingual text gathering can be a problem. This issue, called data sparseness, can significantly downgrade the performance of the system. This work aims at alleviating this problem.

Generic methods

An additional goal is to develop methods that do not make assumptions about the languages involved, or the kind of input. While performance might benefit more from using language specific tools and techniques, one of the advantages of statistical machine translation is that it allows works on an abstract level, without specific reference to language pair. The methods described here are not designed with a particular language in mind, thus making it possible to be used on a wide variety of language pairs. In addition to that, the way linguistic knowledge is incorporated is quite simple and does not require any special annotation of the text. This makes easy the modifications to the current scheme, for example changing the way the morphologic analysis is done. Finally, the current scheme just relies on the availability of parallel text, which is eitherway a prerequisite to start developing an SMT system. There is no need for language specific tools like parsers. This is especially important because while there has been a great deal of research about a few languages (like English, French), resulting in a variety of tools, for the under resourced languages very few tools exist. It would not make sense to build a system that instead of relying on large parallel corpora relies

on elaborate linguistic analysis, when none of these exist for a language.

Chapter 3

Baseline system

3.1 Theoretical background

The baseline system is a phrase based statistical machine translating system, translating from Greek into English and vice versa, described in detail in [14]. It tries to find the most probable native sentence $e_1^I = e_1 \dots e_i \dots e_I$, given the foreign sentence $f_1^I = f_1 \dots f_j \dots f_I$. Searching among all[1] the possible translations the most probable one is chosen:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I} \{Pr(e_1^I|f_1^J)\} = \tag{3.1}$$

$$= \arg\max_{I,e_1^I} \{Pr(e_1^I)Pr(f_1^J|e_1^I)\} \tag{3.2}$$

where the decomposition represents the well known noisy channel approach applied to statistical machine translation by [5]. $Pr(e_1^I)$ is the language model while $Pr(e_1^I|f_1^J)$ is the translation model. The log-linear model proposed by [23] gives

$$Pr(f_1^J|e_1^I) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{I',e'^{I'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J)\right)} \tag{3.3}$$

[1] Actually the search is not exhaustive, that would take too long

where $h_m(e_1^I, f_1^J)$ are the various feature functions for a native/foreign phrase pair (e_1^I, f_1^J), and λ_m the corresponding weights. The denominator in Equation 3.1 is a normalization factor so it can be omitted thus reaching the following form of the decision rule.

$$\hat{e}_1^{\hat{I}} = \arg\max \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{3.4}$$

The weight vector λ_1^M is maximized according to the translation quality, measured by the BLEU metric, using the process described in [22]. Note that the language model is a special case of feature function where we are only concerned with the native language.

The system based on an initial implementation from [30], and its performance is enhanced by elaborating the sentence alignment and phrase extraction parts. Below follows a brief description of the parts used and the complete process from training the system to using it to obtain the translation of an input text.

3.2 Corpus preprocessing

The corpus is not available in a convenient format for machine translation. This Section reviews the preprocessing steps required to bring the texts in the form needed to train the system. The first step is to detect the sentence boundaries in the corpus and after to associate the sentences in the one side of the corpus with their equivalent translations.

3.2.1 Sentence boundary detection

The first preparation step is detecting the sentence boundaries. Sentence boundaries are denoted in text by special characters like *. ? ! ;* . It is not always true, however, that these punctuation marks denote the end of a sentence since they can be part of an abbreviation, as is the case with *i.e.*. To overcome this problem a list of abbreviations was assembled from online dictionaries. Such a list cannot be complete, so a probabilistic mode was applied on the corpus which checks if a token could be abbreviation or honorific. The method was applied on 111408 English and 112756 Greek sentences, and the results were evaluated by a human judge. The results are displayed in Table 3.1. Most mistakes are caused by spelling mistakes in the text.

	Greek	English
Sentences	112756	111408
Candidate Punctuation Marks	120225	113207
False detections	205	175
Error (%)	0.17%	0.15%

Table 3.1: Sentence boundary detection statistics

3.2.2 Sentence aligning

Having detected the boundaries of each sentence, it is now possible to establish the correspondence between the two sides of the parallel text. This part is quite important, because low quality will hamper the performance of the system. It is also important to come up with an unsupervised solution to this problem because parallel corpora can be quite a hundred thousand

sentences long and it would require a lot of human work hours to manually identify correspondence between the languages involved.

This process, called Bilingual Sentences Alignment, relies on the assumption that there is a correlation between the sizes of corresponding sentences. It is an implementation of the Church & Gale algorithm, presented in [9]. The results for a set of 2051 aligned sentences are shown in Table 3.2.

Category	total count	mistake count	Percentage of errors
1-1	1912	19	0.99
1-0 or 0-1	3	3	100
1-2 or 2-1	139	9	6.1

Table 3.2: Bilingual sentence alignment evaluation

Category **1-1** corresponds to one English sentence being translated into one Greek sentence, **1-0** and **0-1** one English sentence not being translated into a Greek one and vice versa, while **1-2** and **2-1** to two sentences in one languages to being translated into one in the other language.

3.3 Word alignment

Having now established a correspondence between the translated sentences in the parallel corpus, we need to find out in each phrase pair which chunks of words are translated into which. This process should be unsupervised, that is, no human intervention should be necessary to annotate the sentences beforehand. Although this seems quite difficult to achieve there is a simple solution that works quite well. We begin by assuming all alignments to be equiprobable, and then use the Expectation Maximization algorithm to find

out the most probable. This is an iterative process, which is implemented by the freely available toolkit GIZA++ [23]. One problem is that the GIZA++ toolkit produces alignments with the inherent constraint that only 1 word in the source language can be aligned to N words in the foreign language. By intuition this is not correct, as context information is important, changing the meaning of single words. The solution used is to run the process bidirectionally, using each language as source and target.

3.4 Translation table building

The next step is to build the translation table. The translation table consists of entries of the form :

source phrase	target phrase	feature scores

To construct this table first we need a set of bilingual phrases, which are extracted from the GIZA++ alignment. For a sentence pair (e_1^I, f_1^J) where I is the length of the source sentence and J of the target one the alignment produced by GIZA++ can be viewed as an IxJ matrix A. The bilingual phrases are the ones which satisfy the following criterion:

$$BP(f_1^J, e_1^I, A) = \left\{ \left(f_j^{j+m}, e_i^{m+n} \right) : \forall (i^{'}, j^{'}) \in A : j \leq j^{'} \leq j+m \leftrightarrow i \leq i^{'} \leq i+n \right\} \tag{3.5}$$

In this system the maximum length of the bilingual phrases is 4, so in Equation 3.5 the additional constrained $m, n \leq 4$ is imposed. This constrain is used because bigger sizes lead to very big translation tables, which take long time to train and occupy more space. An additional problem is that their handling becomes more cumbersome. In addition to that, in [27] it has

been experimentally established that incorporation of higher order ngrams does not improve significantly performance[2]. The baseline system uses a set of five features for each bilingual phrase. Namely,

Bidirectional phrase translation model The probabilities are approximated using the Maximum Likelihood estimation:

$$P(\bar{f}|\bar{e}) \approx P_{ML}(\bar{f}|\bar{e}) = \frac{c(\bar{f},\bar{e})}{\sum_{\bar{f}} c(\bar{f},\bar{e})} \tag{3.6}$$

It is the most important feature of the translation table and it models how probable is to observe phrase e as a translation of phrase f and vice versa.

Word penalty model The word penalty is used to control the length of the produced translation, end is computed using the following equation:

$$P_{pen}(f_1^J, e_1^I) = I \tag{3.7}$$

The reason to include this model is to control the tendency of machine translation systems to produce short translations.

Distortion model The distortion model is actually computed at decode time, since it is dependent on the order at which the phrases are translated. It discourages the decoder to change the order in which the phrases are translated. It is convenient to use when translating between highly correlated languages, as is the case with Latin languages, also because it prunes the search depth. When dealing however with

[2]This is not necessarily true when training a system on a specific domain, where the vocabulary will be quite small; in this case it might make sense to increase the order of ngrams

highly different languages pairs, as from Japanese into English it is necessary to relax this constraint, since in this case there are profound differences in syntax. It is computed using the following equation:

$$D(e, f) = -\sum_{i} d_i \tag{3.8}$$

where d_i for each target phrase e_i produced is computed as $d_i = |a_i - b_{i-1} + 1|$, where a_i is the first word position of the ith translated phrase and b_{i-1} is the last word position of the $(i-1)$th translated phrase.

Language model The language model computation is described in the next section. It serves as a way to ensure that the sentences produced are fluent, and not just a meaningless concatenation of phrase chunks.

3.5 Language modeling

Language modeling is an integral part of the translation process. It is the main way to model the fluency of the produced translation. The most elementary, yet surprisingly effective form is ngram language model. The idea is that the probability of an I length sentence e_1^I is computed as the product of the conditional probabilities of each word on the previous n

$$P(e_1^I) = \prod_{i=1}^{I} P(e_i | e_{i-n+1}^{i-1}) \tag{3.9}$$

What is needed in Equation 3.9 is an efficient way to compute the probabilities $P(e_i | e_{i-n+1}^{i-1})$. The simplest way is to compute the Maximum Likelihood estimation of this term,

$$P(e_i | e_{i-n+1}^{i-1}) = \frac{c(e_{i-n+1}^{i})}{c(e_{i-n+1}^{i-1})} \tag{3.10}$$

where $c(e_a^b)$ denotes the count of the string $e_a \dots e_b$.

The counts are calculated using a sufficiently big corpus. How big is sufficient cannot be easily answered. The first problem arising is that for a 10000 word vocabulary the possible 4grams are 10^{12}, and such a big corpus is very difficult to gather. Even if it is available, the implementation problems of gathering the counts and computing the probabilities pose a big problem. One can ofcourse argue that the biggest part of these permutations are noise, phrases syntactically incorrect without ant meaning. Even though, the problem is still apparent. An additional problem is that ML works well when the number of observations is high, so even for phrases that exist only a few times the probabilities estimated will not be reliable. There has been a great deal of research on this area. One way to deal with this problem is discounting, where the probabilities of well observed phrases are redistributed among the less frequently seen/unseen ones. All these techniques are implemented in the freely available toolkit SRILM [32]. For the baseline system a 4gram language model is trained using the Chen and Goodman's modified Kneser-Ney discounting described in [6].

3.6 Decoder

Having trained all the necessary models, it is now possible to translate a source sentence into the target language. What is needed is to search among all the possible target language sentences and choose according to the criterion of Equation 3.1. The search is conducted using the Moses decoder, described in [17]. Moses is a replacement for the Pharaoh [15] decoder. Its main differences is that it is open source, thus allowing the modification of

the search process and that it offers the possibility of performing factor based translation, which is a generalization of phrase based translation where each word is replaced by a vector with arbitrary entries. Moses is currently widely used in the research community.

Moses implements a beam search. Instead of searching exhaustively among all possible translations, it limits the search space to only a radius around the best translation found so far. It should be noted that the algorithm used can be also used to perform exhaustive search, but the small performance improvement does not justify the increased time and space complexity. The user can modify various parameters to tune for quality or speed; it is possible to change the translation table size, the hypothesis stack size (the beam of the search) and the reordering limit. In addition to that it is possible to change the weights that are assigned to the translation and language models. An interesting feature is that it is possible to get the n best translations for an input sentence. It is then possible to perform rescoring on this list by adding features that are too expensive to incorporate directly into the decoder and this way improve the translation quality. Another interesting feature is that it supports factored based translation. Words are replaced by vectors which might contain the word itself, the Part Of Speech tag, semantic labels and other kinds of information. This is an interesting option, because it allows for integration of linguistic information into the translation process. It should be noted, however, that in this case the translation process is much more expensive requiring much more time and computational resources.

Chapter 4

Corpus

4.1 Introduction

As already mentioned, the corpus used are the Proceedings of the European Parliament [16]. It has been chosen because it is available on a large number of languages, thus allowing for easily testing the performance of the system on different language pairs. It is also widely used in the community of statistical machine translation, allowing for comparable results with other approaches. In this Chapter the characteristics of the corpus for the Greek - English pair are analyzed.

4.2 Corpus statistics

The characteristics of the corpus are depicted in Table 4.1. For each language the whole 540k sentence corpus is used as well as four subsets.

For the two languages and the different sets, are counted the

- number of word forms (distinct words appearing in the text)
- of tokens

- of singletons (word forms appearing only once)
- of not observed word forms in a 5000 sentence test set (Table 4.2)

A first observation concerns the number of words appearing only once is a substantial part of the corpus. For example, in the 40k sentence Greek corpus, 16956 out of the 39110 total words appear only once. One can accurately predict that these words will not be correctly translated. The number of not observed words is also quit high, 1195 for English and 2450 for Greek. Although increasing the corpus size lowers the number of not observed word forms, the same does not apply for singletons, which number steadily to one third of the number of word forms. It is also interesting to note an important difference between the two languages. While the number of tokens is almost the same, the number of Greek word forms is bigger than the equivalent figure for English by a factor of 2-2.5, depending on the corpus size. This accounts for the fact that the number of singletons and not observed words in Greek are more than in the English text by the same factor, more or less.

In addition, in Figure 4.1 one can see the distribution of word frequencies in the 40k sentence corpus. In the horizontal axis are the frequencies of appearance of a word in the corpus, and on the vertical the size of a group of words that have the same frequency. Inspection of this figure reveals that most words only appear in the corpus a few times. For the English side, the number of words that appear more than 10 times is just 4760, a mere 24% of the corpus, while the equivalent percentage for Greek is 17%. Considering that translating is more than just using a dictionary mapping of individual words, but context plays an equally important role, it becomes pretty obvious that the majority of the words will lack sufficient context information to build

translation rules.

4.3 Conclusion

As the above statistics suggest, training a system using the Europarl can be a quite challenging process. The low frequency of appearances of words can be attributed to two reasons:

morphology The richer the morphology of a language the bigger the number of word forms. In the Greek-English example this is obvious, where the ratio of word forms per tokens is 0.0225 for English while the same figure for Greek is 0.0448, more than double.

domain the nature of the text; unlike specific domain corpora, the Europarl is a transcription of the proceedings of the European parliament. As a result the topic of the conversation is not constrained, but can include politics, economic, military as well as other subjects to a smaller extent. This essentially means that for sure the test corpus will contain words that have not been observed in the train corpus enough times, or even at all.

Another problem, not obvious from the previous analysis, is that the proceedings are not really translations in the strict sense, but rather rendering. This can pose a significant obstacle in identifying improvements, because we are constrained to using only one available reference translation to evaluate the system output. This can prove quite a hindrance, because in the case where an improved translation is using different words to convey a meaning

equivalent to the reference translation, the current evaluation techniques will not detect the improvement over a totally wrong translation.

language	type	sentences	word forms	tokens	singletons	not observed
English	normal	40k	19858	882003	7396	1195
	normal	80k	26574	1771690	9533	814
	normal	160k	36380	3544252	13329	518
	normal	320k	50241	7065540	18736	361
	normal	540k	61820	11933322	23543	278
English	stemmed	40k	13534	882003	4892	881
	stemmed	80k	18155	1771690	6463	647
	stemmed	160k	25253	3544252	9516	469
	stemmed	320k	37155	7065540	15373	344
	stemmed	540k	48310	11933322	21045	274
English	split	40k	13534	882003	4892	881
	split	80k	18256	2461856	6456	645
	split	160k	25348	4925518	9510	468
	split	320k	37244	9811677	15362	344
	split	540k	48393	16554523	21033	274
Greek	normal	40k	39110	875577	16956	2450
	normal	80k	54078	1751859	22335	1659
	normal	160k	73687	3485946	29395	1047
	normal	320k	115910	6981613	49760	645
	normal	540k	155337	11836340	67838	492
Greek	stemmed	40k	23278	875577	8625	1534
	stemmed	80k	30858	1751859	10827	1115
	stemmed	160k	40742	3485946	14082	846
	stemmed	320k	74482	6981613	34766	603
	stemmed	540k	112632	11836340	56914	471
Greek	split	40k	23693	1370565	8653	1537
	split	80k	31284	2741915	10832	1115
	split	160k	41165	5454721	14075	846
	split	320k	74898	10879833	34748	603
	split	540k	113033	18381187	56888	471

Table 4.1: Train set characteristics (40k-80k-160k-320k-540k sentences)

language	type	word forms	tokens
English	normal	8151	121166
	stemmed	5566	121166
	split	5760	174591
Greek	normal	13487	117607
	stemmed	8415	117607
	split	8900	184413

Table 4.2: Test set characteristics (5k sentences)

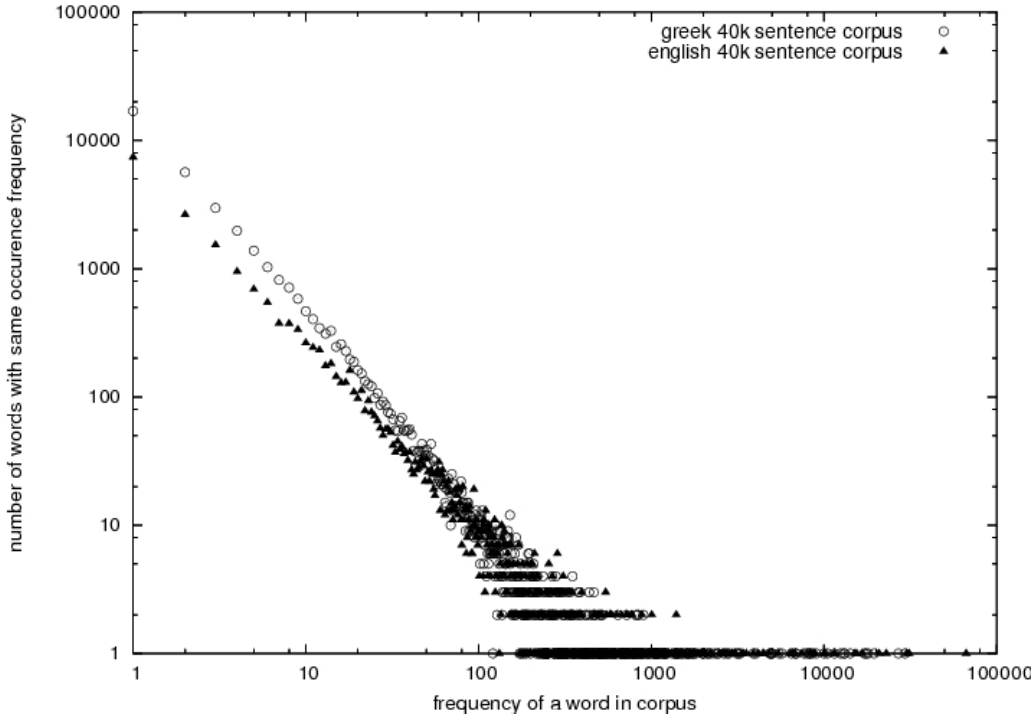

Figure 4.1: Frequencies of words in 40k sentence corpus

Chapter 5

Morphology

5.1 Introduction

It is obvious from the above that a translation system trained on the above corpus will display poor performance since many words will first time be observed in the test corpus. One way to deal with this problem is to use more data for training, which however, is not always an option since parallel texts are not in abundant supply. Another one is to better exploit the available parallel corpora and also to use monolingual corpora, which are much easier to find, so as to address the data sparseness problem.

In natural languages, instead of using a totally different word for each and every possible meaning, words that convey similar meaning are similar themselves, usually differing in some parts of them (e.g. their endings). This is the basic concept that constitutes the notion of the morphology of a natural language.

Since morphology is an essential structural element of a natural language, it comes as no surprise that it is already widely used in Natural Language Processing applications. In Information Retrieval, for example, the notion

of stemming is common, which refers to the segmentation of a word into root (stem) and suffix. In Statistical Machine Translation, however, there is need for more complex representation, modelling the interaction between morphemes and groups of morphemes.

The rest of this chapter is organised as follows. Section 5.2 is devoted to exploring what is a morphology. Section 5.3 reviews methods for arriving in a morphologic analysis of a natural language. Section 5.4 describes the way morphological analysis is extracted from raw monolingual corpus for this work. Section 5.5 deals with a small postprocessing step applied on the analysis derived from Linguistica in order to improve its precision. Section 5.6 discusses the impact of morphological analysis on the characteristics of the corpus used in the experiments done.

5.2 Natural language morphology

5.2.1 Morphemes and the kinds of morphologies

Morphemes are defined as the minimal meaning-bearing units in a language. Apart from the stem of a word, a morpheme can be an affix, which usually provides additional meaning of some kind to the main concept that is provided by the stem. An affix may be a prefix, suffix, circumfix or infix, whether it precedes the stem, follows it, does both or is being inserted in it, accordingly. Prefixes and suffixes (and circumfixes as well, since they may be viewed as a combination of a prefix and a suffix) are often called concatenative morphology, since a word is composed of a number of morphemes concatenated together. In some languages, morphemes are combined in complex ways, using what is called nonconcatenative morphology. Another kind

of this type is the templatic morphology that is very common in languages like Arabic, Hebrew etc. and uses root words and templates that transform them

There are two broad classes of ways to form words from morphemes: inflection and derivation and thus we speak of inflectional or derivational morphology. These two are partially overlapping, since the borders between them are usually not absolutely clear. Inflection mostly deals with the usage of affixes, while derivation is the combination of a word stem with a grammatical morpheme usually resulting in a word of a different class, often with a meaning hard to predict exactly.

Three general classes of linguistic knowledge are needed in order to build a morphological parser:

Lexicon The list of stems and affixes, together with basic information about them.

Morphotactics The model of morpheme ordering that explains which classes of morphemes can follow the other classes of morphemes inside a word.

Orthographic rules Spelling changes that occur due to morpheme attachment.

5.2.2 Learning a morphology

In recent years, there has been much interest in computational models that learn aspects of the morphology of a natural language from raw or structured data. These models are of great practical interest, minimizing the expert resources or need of linguistics in order to develop stemmers and analyzers.

There are three distinct ways of learning a language' s morphology:

Supervised learning The data consists of a set of pair of words.

Unsupervised learning The data consists of a single set of all the words in the corpus.

Partially supervised learning The data consists of two sets of words, without any indication of the relationship between the individual words.

We will mostly deal with unsupervised learning, since such methods may be used with untagged corpus which is often the case, performing morphological analysis based only on a corpus. This can be a valuable tool that may be used in statistical machine translation, where the system is being trained using such untagged corpora.

5.3 Acquiring a natural language morphology

5.3.1 Introduction

In this section, the most important approaches of (mostly) unsupervised morphology learning are presented. One way to categorize the existing approaches on this matter is by evaluating whether human input is provided in the process of deriving the morphology and whether the goal is to only obtain affixes or to perform a complete morphological analysis. According to this categorization, we may therefore cluster the various approaches and techniques as follows:

- Bootstrapping using a knowledge source

- Obtaining affix inventories
- Performing a complete morphological analysis

For the first two categories we will provide short descriptions, while for the third one we will describe in detail an example application.

5.3.2 Bootstrapping using a knowledge source

A first approach in obtaining morphologies is to begin with some initial human-labeled source from which to induce other morphological components. Although their work may be more suited to information retrieval (IR), Xu and Croft[33] are proposing a technique that is an example to this case. They are basing their work around the hypothesis that the word forms that should be conflated for a given corpus will co-occur in documents from that corpus. They use a co-occurrence measure to modify an initial set of conflation classes generated by a stemmer, refining the output of the well known Porter stemmer. This corpus-based stemming automatically modifies the equivalence classes (conflation sets) to suit the characteristics of a given text corpus. They perform experiments in English and Spanish, but they do agree that generating the initial conflation classes in languages with more complex morphologies may be a problem.

5.3.3 Obtaining affix inventories

A second, knowledge free category of research has focused on obtaining affix inventories. DeJean[8] is inspired by the works of Zellig Harris[13], a distributional approach where the distribution of an element is the set of the

environments in which it occurs. His work uses untagged and non artificial corpora without specific knowledge about the studied language. The algorithm is divided into three steps: the first step computes the list of the most frequent morphemes, which is being extended in the second step by segmenting words with the help of the morphemes already generated, while the third step consists in the segmentation of all the words with the morphemes obtained at the second step. A symmetric procedure can be used to identify prefixes; the letters of the words are just reversed. Morpheme boundaries for the most frequent morphemes are discovered when the number of different letters that are found to follow some sequence of letters is higher than a threshold.

5.4 Linguistica

5.4.1 Introduction

Linguistica [10], is a freely available for research usage toolkit which performs a complete morphologic analysis of a natural language, relying on the Minimum Description Length model. The process of training is fully unsupervised, so it provides an easy way to come up with a morphologic analysis for a new language, provided that there is a corpus available. This characteristic is very important, as it does not constrain the methods explored in this work to a specific language pair; if one wants to apply them in a different one, it is not necessary to find tools to perform morphologic analysis, but just to use Linguistica on the new corpus. This Section start with a brief review of the MDL model, continues with a description of Linguistica and concludes with a simple heuristic rule used to postprocess the resulting

analysis in order to improve the accuracy.

5.4.2 Minimum Description Length Model

The central idea of minimum description length (MDL) analysis[29] is composed of four parts:

1. A model of a set of data assigns a probability distribution to the sample space from which the data is assumed to be drawn.

2. The model can then be used to assign a compress length to the data, using familiar information-theoretic notions.

3. The model can itself be assigned a length.

4. The optimal analysis of the data is the one for which the sum of the length of the compressed data and the length of the model is the smallest.

In other words, we seek a minimally compact specification for both the model and the data. Linguistica tries to analyze words into morphemes, using MDL as guideline. In order to provide a morphology to evaluate using MDL, first bootstrapping heuristics are needed that provide an initial morphology.

5.4.3 Heuristics for word segmentation

Two heuristics are used to produce an initial morphology analysis.

- The first one (called take-all-splits), considers for each word of length of length l all the possible cuts into $w_{1,i} + wi + 1, l,\ 1 \leq i < l$. For each

cut

$$H(w_{1,i}, w_{i+1,l}) = -(i \log freq(stem = w_{1,i}) + (1-i) \log freq(suffix = w_{i+1,l})) \tag{5.1}$$

is computed; then it is used in the following formula to assign a probability to the cut of w into $w_{1,i} + wi + 1, l$.

$$prob(w = w_{1,i} + wi + 1, l) = \frac{1}{Z} e^{-H(w_{1,i}+wi+1,l)} \tag{5.2}$$

where

$$Z = \sum_{i=1}^{n-1} H(w_{1,i} + wi + 1, l) \tag{5.3}$$

For each word the best parse is noted, and then we iterate until no word changes, which typical takes less than five iterations.

- Using the convention that each word ends with an end-of-word symbol we compute the counts of all n-counts between two and six letters (including the end of word). Then for each ngram $[n_1 n_1 \ldots n_k]$ we compute

$$\frac{[n_1 n_1 \ldots n_k]}{total\ count\ of ngrams} \log \frac{[n_1 n_2 \ldots n_k]}{[n_1][n_2] \ldots [n_k]'} \tag{5.4}$$

  The top 100 ngrams on the basis of this measure are chosen as candidate suffixes. Then all words are parsed into stem plus suffix, if possible, using a parse from the candidate set. For those words that more than one parsing are possible, we keep the most probable, according to the previous heuristic.

Consequently, for each stem we make a list of all the suffixes which appear with it, called a signature. Stems having the same signatures are merged.

Initially all signatures with only one stem (which account for about 90% of the initial signatures) are removed, as well as those with only one stem. The remaining are called regular signatures. The resulting signatures are of the form

$$\left\{ \begin{array}{l} stem_1 \\ stem_2 \\ stem_3 \end{array} \right\} \left\{ \begin{array}{l} suffix_1 \\ suffix_2 \end{array} \right\} \tag{5.5}$$

Variations of the resulting grammar are considered and adopted only if they reduce the description length of the grammar and the corpus. First each suffix is tested to see if it is a concatenation of two independent suffixes. Then suffixes in the same signature are tested to see if they begin with the same letter or sequence of letters, so that these letters can be considered part of the preceding stems. Finally signatures with only a small number of stems are checked to see if they are worth keeping, or discarding them leads to a better model.

5.5 Postprocessing of Linguistica analysis

5.5.1 Introduction

Using Linguistica on a corpus we can have a morphological analysis. It is possible, however, especially when the available corpus is small that the produced morphology will not be very good, both in terms of precision and recall. Linguistica offers the chance of adjusting some parameters, to influence the resulting morphology. However, to use them one must take into account the way the morphology is built. We have tried to offer a simple and cheap (both in terms of time and computational power needed) way of in-

creasing the precision of the resulting morphological analysis, on the expense of recall.

5.5.2 Heuristic proposed

Examination of the resulting morphological analysis provided by Linguistica easily leads to an observation. In most words mistakenly analyzed, the error is assigning stem characters to the suffix. The opposite error, assigning suffix characters to the stem is not so important since it is more easy to identify a stem with extra characters in the end than a chopped stem. The problem of false identification is even more important when dealing with short words, where removal of the suffix usually leaves a very short stem (maybe two or three characters long), which is possibly useless for training a statistical machine translation system. To overcome these problems we use a heuristic rule which uses two parameters

- The length of the words l.
- The ratio r of the length of the suffix divided by the length of the whole word.

We examine every word analyzed by Linguistica. We adopt the analysis only for words that have $l_{word} > l_0$ and $r_{word} < r_0$, or else discard it.

5.5.3 Results-evaluation

In order to be able to choose values for r_0 and l_0 we carried out a simple experiment. We used Lingustica to provide morphological analysis based on a 1M token Greek corpus. Then we randomly picked 1k words (2k tokens)

for which Linguistica had produced morphological analysis. To evaluate the performance of the heuristic, a human judge decided for each word if the analysis was correct or mistaken. The results, using different values for r_0 and l_0 are shown in Table 5.1.

r_0	l_0	Precision(%)
1	0	79
0.2	0	89
0.2	4	89
0.2	5	89
0.2	6	93
0.3	0	84
0.3	4	84
0.3	5	90
0.3	6	94

Table 5.1: Linguistica precision for the Greek corpus

5.6 Morphological analysis on Europarl

The results of applying morphological analysis to the europarl corpus can be seen on Table 4.1, in page 25.

One can easily note that the number of singletons and not observed word forms is substantially smaller in both the stemmed and split level than in the lexical one.

The behavior of other tools (TextPro[2], Porter's[1] stemmer[1], Orphano's[25] lemmatizer) is shown in Figures 5.1, 5.2, for the greek and english sides of the corpus respectively. In the horizontal axis are the number of words that

[1]Subset is the subset of rules that only split words, rather than changing letters

can be created from a given stem, while on the vertical the number of stems with the same count of possible word creations. *Linguistica heuristic* is the analysis provided by Linguistica when applying the heuristics rules discussed before. *Porter subset* is the analysis provided from porter stemmer, when using only the subset if rules that do not modify the stem of the word. The graphs show that Linguistica displays comparable performance with other tools. It should be noted that Orphano's lemmatizer is not directly comparable to Linguistica, since it is not a stemmer, but provides the lemma from which a word is derived, thus the difference in behavior between the two tools.

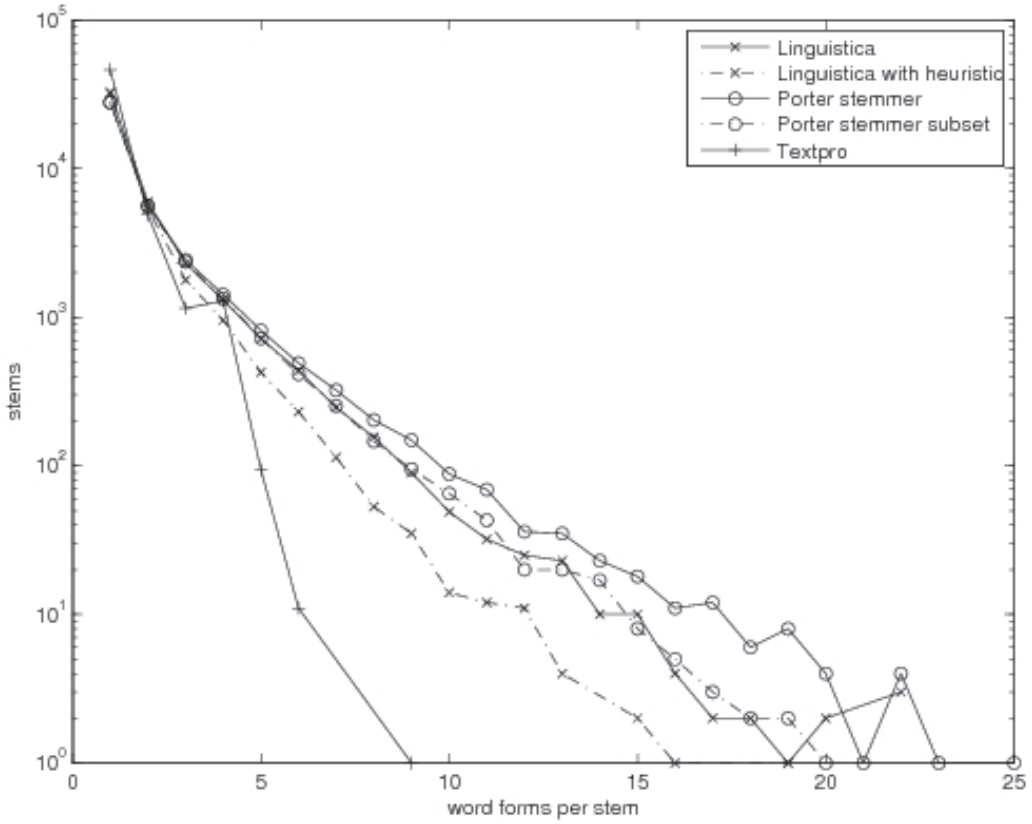

Figure 5.1: Stem count distribution for English using different analysis tools

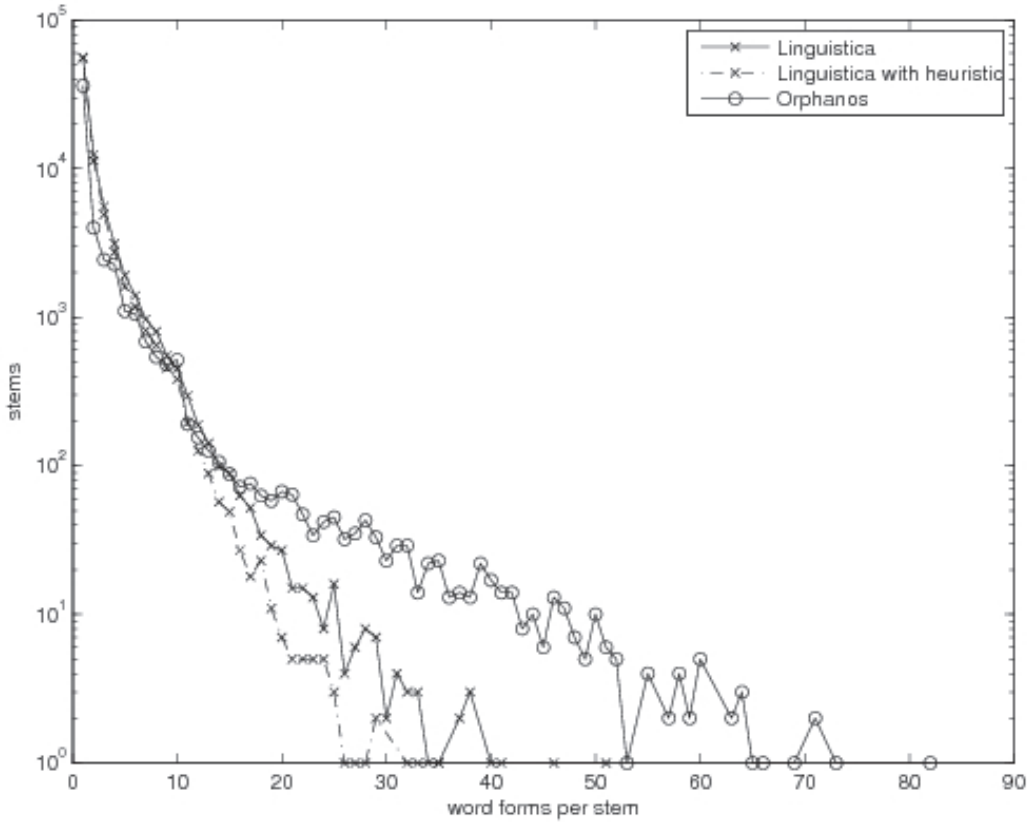

Figure 5.2: Stem count distribution for Greek using different analysis tools

Chapter 6

Morphology incorporation

6.1 Introduction

Having analyzed how a morphological analysis for a natural language can be achieved, it is now time to investigate approaches of incorporating this information into the baseline translation system.

This chapter is devoted to the exploration of different approaches to this problem. There has been some work on this field, on various language pairs. What exactly is considered morphology and how it is incorporated to the base system, however, varies quite a lot. In this work morphology is used in its simplest form; words are just analyzed into stem and suffix or not at all. This is quite a simplistic approach in comparison to related work, where various morphosyntactic tags are used to describe the alterations applied to a base form to produce the word, as seen in the text. In the experiments which follow morphology has been either applied as a preprocessing step, and in some case also as postprocessing. It has also been tried to change the way the translation table is created, in which case the input text is not necessary to be altered. In addition to that, the possibility of using the nbest

list produced by the decoder to improve the translation has been explored. Experiments have been conveyed to combine the output of different systems as well as to assess the upper limit of the quality of the produced translation.

The rest of this chapter is organized as follows; Section 6.2 reviews the relative work in the field. Section 6.3 describes initial experiments conveyed, pre/postprocessing the input and output of the translation system. Section 6.4 extends the ideas of the previous section by incorporating nbest list processing. Section 6.5 describes the main part of the work done, using morphology in order to enhance the translation table coverage. Section 6.6 describes a simple way to combine different systems in the translation table level. Section 6.7.2 deals with combining the output of different systems based on the confidence of each one. Finally, in Section 6.7.3 system combination is performed on the basis of which system is expected to perform better, judging on statistics gathered from the input sentence.

6.2 Previous work

What follows is a review of previous attempts of morphology utilization into SMT systems. The gains in performance reported are relative improvements of the translation quality as measured with the BLEU metric [26].

In [28], the language pairs dealt with are Catalan/Spanish to English and Serbian to English. In the first case a 13k sentence corpus is used, while for the second system the size of the train corpus is just 2k sentences. In the first case only verbs are considered, while for the Serbian to English system all words are treated. For the Spanish/Catalan system syntax information is used in addition. In both systems morphology is used only on the source

side. Linguistic information is used in the first case to split words into the base form (lemma or stem). For the second system only the stem is kept while the suffix is discarded. The improvements reported between 5% and 8%.

In [4], translation is from Spanish into Chinese and vice versa. The train corpus is 28k sentences. Morphology is applied on the Spanish only side of the corpus, which is either stemmed or lemmatized. In the Spanish to Chinese system just the input is analyzed. In the opposite direction, however, a Chinese to analyzed Spanish system is cascaded with an analyzed Spanish to Spanish system. Small improvements are reported.

In [34], systems translating from German and Finnish into English are trained on corpora 5k to 750k sentences. The idea is that the translation table of the baseline system is enhanced with backoff probabilities for the cases where a word has not been observed in the lexical level, but only in the stem level. Improvements up to 8.5% are reported.

In [11] translation is from Czech into English. The corpus is 21k sentences. Source words are analyzed and different representation schemes are tried. The most interesting part of this work is that morphemes are also used to establish the word to word correspondence, which generally is based only on lexical level information. Improvements up to 20% are reported.

In [35], Arabic is translated into English, using a 20k sentence train corpus. The Arabic side of the corpus is split into morphemes, taking into account the probability of each possible split, as well as discarding morphemes which carry linguistic information not present in the English translation. Improvements up to 5% are reported.

In [18], translation is carried out from Arabic into English using varying corpora sizes for training, from 3.5k to 3.3M sentences. Both sides of the corpus are tagged with part of speech tags, and various models are used to find out if each tag in the Arabic corpus is corresponding to an English one, or it can be deleted/merged. This process is also repeated on the to be translated input text. Improvement up to 150% is reported.

In [20], experiments are carried out from English to Russian and Arabic, using corpora of 1M and 500k sentences respectively. The idea is to predict the generation of each target word from stem into the lexical level, depending on features such as the inflections of the context words, the tags of the associated words in the input. The experimental framework, however, does not provide a testing of these method since their performance is only evaluated against an already translated corpus in terms of accuracy.

In [7], morphology is used to improve the word alignment of a 100k sentence German-English corpus. Words in both sides of the corpus are replaced into their citation form and consequently precision and recall improvements on the word alignment are reported. No mention, however, is made of whether this improved alignment actually results in training an improved translation system.

In [21], translation is from German into English using a corpus of 58k sentences. Using morphology words are decomposed to a lemma-tag representation, and various restructuring schemes are used, like merging German verbs with the detached prefixes, idiomatic multiword expression into a single tokens. Performance improvements up to 30% are reported for the resulting translation quality.

6.3 Pre/Postprocessing incorporation

The simplest way to utilize morphological information is as a post/preprocessing step. The advantage is that the baseline system can be used without modifications, since all modifications are done at earlier or later step. These experiments can also be used to assess what kind of improvement can be expected by incorporating higher level linguistic information in the development of the system. Below follows a description of the systems that have been built and evaluated.

- **normal** is the baseline system.
- **stemsource** is the baseline trained with the source language stemmed and the target language same as in the baseline. The motivation is that some morphological aspects of the one language are not translated into the other. Using a more simple form of the word, possibly redundant information is discarded and the result can be improved quality translation. The fact the translation output is in the lexical level helps to avoid implementing a generative model.
- **splitsource** is the baseline trained with the source language split and the target language same as in the baseline. The motivation is same as above, only now the suffixes are not discarded, but kept as individual words. The idea behind building this model is that maybe ignoring all the suffixes in the source language discards useful information. By keeping them it is possible that the ones that are useful will be aligned in the target language, while the ones will not be consistently aligned, thus resulting in low probability translation rules.

- **split** is the baseline trained with both languages split and then, as a postprocesing step, the output is concatenated. The idea is that splitting words into stem and suffix might result in a mapping between stems in the source and target language as well as suffix mapping. Since the translation produced is not only words, but also stems and affixes, the consecutive words in the output are checked against the morphological analysis used to find out if their concatenation corresponds to a legitimate word.

- **stem** is the translation when the system is trained at stem level. The reason this model is built is to check if its performance is better than the output of the baseline system, on a stem basis. If so, it is an indication that there is room for improvement by operating on other than the lexical level.

- **normal_stemmed** is the stemmed output of the baseline system. This model is just used in order to compare with the performance of the **stem** system.

- **normal_giza** is the translation produced by the baseline system using an improved word alignment, trained with stems instead of words. This is done to inspect if the alignment produced at stem level is superior to the one created at lexical level.

6.4 split + rescoring

Experimental results suggest that the split model generates a lot of spurious words. These words are mostly suffixes, and to a less extend stems, which

are reordered during the translation, and therefore it is not possible to concatenate them with a stem during postprocessing. It is also difficult to spot them, because these morphemes can be either words or suffixes. Consider for example the word **on**. It is possible that it is a preposition, or the suffix created by splitting, for example, **decision** into **decisi + on**.

One approach to deal with this ambiguity is to tag all suffixes created with a special token. That way, identifying spurious suffixes becomes straightforward. Since the translation process, however, results in reordering of the tokens in the output, it is possible that even though it is possible to identify a token as spurious, it is not following a stem in order to concatenate them in to a legitimate word. To overcome this problem, we apply this technique on the nbest list provided by the decoder. Starting from the first entry we traverse the list downwards and pick as translation the first hypothesis encountered which does not contain a spurious suffix. The process is described below.

- Source text split as usually (decision → decisi on)
- Target text split with special character (decision → decisi **X**on)
- 4gram language model built
- After translation, keep 1000-best list [1].
- Try to merge words. (decisi **X**on → decision)
- Find best scoring translation without **X** symbol

[1] Actually smaller list size does not harm performance, because low scoring sentences are eitherway bad translations

6.5 Generating translation rules

One different approach is to train a stem to stem system and then try to generate rules consisting of words. The idea is to first train a stem to stem translation system, for which the corpora available are less affected by data sparseness, and then to transform the stem level rules into lexical level rules. This is based on the assumption that high probable lexical level rules can de derived by correspondingly high level stem level rules. So what is needed is to provide a mapping from the stem level rules to lexical level. Such a mapping ofcourse will have to take into account the probability of the derivation of a certain word from a stem. The approach consists of the following steps

1. Stem corpus using the analysis provided by Linguistica
2. Train stem to stem translation system
3. For each rule produced (stem level) generate the possible rules created (word level)
4. Rank the produced rules and choose top ranking

The implementation of the first two steps is straightforward and already described above. What is needed is to create a mapping from rules in the stem level into rules in the lexical level.

$$P(e|f) = F\left(P(e_s|f_s)\right) \tag{6.1}$$

where e and f are the native and foreign sides of a translation rule in lexical level, and e_s and f_s the native and foreign sides of a translation rule

in the stem level. This process is approximated by the following formula:

$$P(e|f) = \alpha P(e_s|f_s) \exp\{\lambda_1 P(e|e_s) + \lambda_2 P(e_a) + \lambda_3 P(f_a) + \lambda_4 P(e_a|f_a)\} \tag{6.2}$$

where α is a normalization constant, $P(e|e_s)$ the probability of deriving the lexical rule e from the stem level rule e_s, e_a and f_a is the sequence of suffixes in the e and f phrases respectively, $P(e_a)$ and $P(f_a)$ the corresponding language modes and $P(e_a|f_a)$ the probability of observing the suffix sequence e_a given f_a in a translation rule $e|f$. Intuitively Equation 6.2 states that the probability of a lexical level rule is analogous to the corresponding stem level rule. $P(e|e_s)$ models how probable is to generate e from e_s, based on the stem to word probability. The terms $P(e_a)$ and $P(f_a)$ constrain the possible generations to the ones whose suffix sequence is syntactically correct and finally $P(e_a|f_a)$ is a measure of how probable is to observe the suffix sequence e_a in a rule $e|f$.

The probability of a word e given a stem e_s is approximated using the Maximum Likelihood estimation

$$P(e|e_s) = P_{ML}(e|e_s) = \frac{count(e, e_s)}{\sum_e count(e, e_s)} \tag{6.3}$$

and the stem to lexical level probability of a lexical sequence $e_{l1}^I = e_{l1} \dots e_{li} \dots e_{lI}$ given the stem sequence $e_{s1}^I = e_{s1} \dots e_{si} \dots e_{sI}$ is calculated as

$$P(e_{l1}^I|e_{s1}^I) = \prod_{i=1}^{I} P(e_{li}|e_{si}) \tag{6.4}$$

Implementation We will now review the calculation of the terms in Equation 6.2. For $P(e_s|f_s)$ we just need to stem the available bilingual corpus using

Linguistica and then use the baseline system to create the translation table, whose entries will be the probabilities we are looking for. The terms $P(e_a)$ and $P(f_a)$ are easily found by processing the available monolingual corpora using Linguistica and discarding the stems of all the words, keeping the affixes and calculating a 4gram language model on the resulting corpus. In case a word is not analyzed, it is simply replaced by a unique token. $P(e_a|f_a)$ is easily obtained by transforming the translation rules of the baseline system and keeping only the suffix of each word. In order to evaluate Equation 6.3 for all the possible stem to word generations we stem the available corpus and calculate the probabilities needed.

In order to efficiently create all the possible mappings of a stem sequence to the lexical sequences finite state machines are used. Each rule in the stem translation table is represented as an acceptor F_s. A transducer $T_{s \to w}$ representing Equation 6.3 is created and the composition $F_s \circ T_{s \to w}$ creates all the possible rules. For the finite state machines the Carmel [12] toolkit was used.

Computational issues The result of the above composition is a graph with all the possible rules. The size of the graph, however, can vary considerably. Given that a typical translation table consists of 500k-800k entries (depending on the max ngram size and the corpus size), the number of possible rules quickly grows out of control; for a 4gram translation table with around 500k stem rules the possible rules produced are 96.08e+12. If one uses 8gram translation table, 770779 rules, the possible rules are an astonishing 16980.92e+12. It is obvious that such a huge file cannot be even stored

in a hard disk, let alone be used as a translation table. The solution to this problem is well known; pruning. For each stem rule only the 1000 most probable are stored, out of which the top ranking ones are picked.

6.6 Translation table level system combination

Since the system proposed in 6.5 is quite different from the baseline, it is worth investigating the combination of these two systems. One way to achieve this is to create a translation model based on the baseline and rule-generation translation models. The basic idea is that for each rule $P_{lex}(e|f)$ present in the baseline system we compute a new score $P(e|f) = aP_{lex}(e|f) + (1 - a)P_{gen}(e|f)$, where $0 \leq \alpha \leq 1$ and $P_{gen}(e|f)$ is the equivalent rule in the generation system which is created using the procedure described in 6.5. The resulting system is referred to as **interpolation**. Another approach tried is not to interpolate the rules, but to use them both, effectively adding two features to the translation table. This system is called **interpolation_4cols**.

It should be noted that the two systems (baseline and rule generation) do not have the same rules. While the rule generation system translation table has around 8101708 rules, the equivalent figure for the baseline system is 460221. Of these rules the 369987 exist also in the rule generation system, while the rest 90234 do not. For the non-existent ones we just keep the value of the baseline system.

6.7 Sentence level combination

6.7.1 Motivation

Up to now we have dealt with modifying the baseline SMT system in order to obtain improved quality translation. A different approach, complementary to this one, is to combine the output of several MT systems which might result in a better performance. There has been extensive work in this field. Regardless of the way the combination is made, all research on this topic agrees on one issue; the systems to be combined must be uncorrelated [19].

Generally speaking there exist two ways to integrate the output of different systems. In both cases the first step is to translate the input independently using all the available systems. For producing the final translation each method uses a different technique.

Sentence level Use some global score functions for each hypotheses and choose, for each input sentence, the best scoring one.

Phrase level Break each hypothesis into phrases and choose between phrases.

In this work we have constrained ourselves to only applying sentence level combination.

6.7.2 System combination using decoder scores

Both the baseline and the improved system provide a score with each translation. The most straightforward way to combine their output is to choose for each source sentence the one produced by the most "confident" system, that is, the system supplying the highest score for the sentence at hand. If

for each source sentence f we call the ith's system translation e_i and the corresponding score $P(e_i)$, the sentence e is chosen among the candidates according to the following decision rule

$$\hat{e} = \arg\max_i P(e_i) \tag{6.5}$$

6.7.3 System combination based on input

The system using morphology information is expected to outperform the baseline system in cases where the input contains words/phrases that have not been observed in the training corpus enough times to train reliable translation rules. This intuition leads to the idea that it is possible to combine their output by choosing the translation from the morphology system when the input contains segments that have not appeared with high frequency in the train corpus. The decision rule used relies on the mean frequency with which the words of the input appear in the train corpus. If this figure is above a threshold the baseline translation is chosen; if not the translation of the rule generative system is chosen.

As seen in Chapter 4 however, there are a few words that appear with great frequency in the corpus. These words need not be taken into account since

1. Good translation rules exist for them anyway

2. Their bigger frequency defines decisively the mean frequency

For each source sentence f_1^I we pick the translation between the baseline and the generative system hypothesis according to the following rule

$$e = \begin{cases} e_{base} & , R \geq R_o \\ e_{gen} & , R < R_o \end{cases} \tag{6.6}$$

where R_o is an arbitrary threshold, and R is computed using the equation 6.7

$$R = \frac{\sum_{i \in A} c(f_i)}{|A|} \tag{6.7}$$

where $c(f_i)$ is the count of word f_i in the train corpus, A a set with the property $c(f_i) \leq T, \forall i \in A$ and $|A|$ its element count. The values of the constants R_o and T are experimentally defined.

Chapter 7

Results

Introduction

In this section are presented the results of experimenting with the previously described techniques. For these experiments the 40k sentence subset of the available corpus has been used for training, because it allows for quicker development cycle and the improvements from these approaches are expected to be more significant with small corpora, where data sparseness is more severe. The experiments in 7.1 and 7.2 were conducted using a 5000 sentences test set, while the rest on a 500 sentence subset of the test set, because the amount of time needed for the latter is much bigger. For the same reason the second group of experiments were only conducted from Greek into English only.

7.1 Pre/Postprocessing incorporation

The results for the methods described in 6.3 are shown in Table 7.1. All systems were evaluated using the BLEU [26]/NIST [24] metrics on the 5000

sentence test set.

We can see that all systems except for **normal_giza** perform worse than the baseline. For **stemsource** and **splitsource** this can be attributed to the fact that the information discarded is more valuable than the expected gains. For **split** one additional reason is the problem of concatenating words in the translation output. The only case where the modified system outperforms the baseline system is **normal_giza**. This can be attributed to the fact that less word forms help produce a better word alignment, resulting in improved translation model. However the difference is too small to be considered satisfactory.

	gr2en		en2gr	
	BLEU	NIST	BLEU	NIST
normal	0.1911	5.8698	0.1389	4.8008
splitsource	0.1855	5.7289	0.1348	4.7176
stemsource	0.1898	5.8274	0.1339	4.6373
split	0.1873	5.7276	0.1343	4.6895
normal_giza	0.1935	5.8835	0.1408	4.8388
normal_stemmed	0.1947	5.9628	0.1475	5.1338
stem	0.1959	6.0060	0.1453	5.0906

Table 7.1: Scores for pre/postprocessing systems translating from Greek into English and vice versa

7.2 split + rescoring

Using the methodology described in 6.4 the results of Table 7.2 are obtained. **normal** is the baseline translation system, **split** is the normal split system, and **split_special** is the system described in 6.4. The items marked sub-

set, correspond to the subset of the translation, where we were able to find a translation in the lattice without spurious suffixes (4156 out of 5000 sentences). This approach does not yield any improvement. It is interesting to note that for the subset that it is possible to find translations without spurious words the translation quiality is ofcourse higher, but the baseline system is still better. This implies that this subset contains sentences that are easier to translate, so no performance is gained by splitting the words.

	BLEU	NIST
normal	0.1911	5.8698
split	0.1873	5.7276
split_special	0.1895	5.7566
normal_subset	0.2029	5.9753
split_subset	0.1991	5.8708
split_special_subset	0.2011	5.8780

Table 7.2: Scores for the split and rescoring system translating from Greek into English

7.3 Generating translation rules

In order to build the translation table of the system described in 6.5 we must first decide how many lexical rules will be generated per stem rule[1]. For this reason we checked the quality of the translation, as measured by the BLEU score, for different values of r. The results are depicted in Figure 7.1. The horizontal axis value corresponds to the maximum number of lexical rules generated per stem rule, while the vertical to the score of the produced translation. The systems in the graph are:

[1]From now on we will refer to this ratio as r

baseline , the system described in Chapter 3

rule generation system , is using the weight vector $\lambda_1^4 = [1110]$ (from Equation 6.2)

rule generation system with affix translation features , is using the weight vector $\lambda_1^4 = [1111]$

For $r < 10$ the results are quite below the baseline system. After they are comparable and the maximum improvement is when $r = 24$. This is the value we have used for the rest of the experiments.

Another interesting observation regards the comparison between the two rule generation systems. While for small values of r the system incorporating affix translation features is clearly better than the simple one, for larger values of r ($r > 5$) the situation is inverted, since the affix incorporating system is quite inferior. This can be attributed to the fact that for small values of r just a few lexical level rules are generated per stem rule, many of which are not syntactically correct, so include the affix translation feature is important. For bigger r values, however, the "good" rules are anyway included, and its inclusion actually harms performance. This does not mean that it is not a useful feature, but rather that it is not correctly modelled at present. Since at its present form the affix translation modeling it does not help, it is not used, and only the *rule generation* system is used in the next experiments.

After that, Minimum Error Training [22] is performed on both the baseline and the generative system, in order to find the optimum weighting scheme for each, in reference to the achieved BLEU score. The results are given on Tables 7.5 and 7.6, in the column *generative*. The systems are tuned on the

development test, while the performance is tested on test sets 1-9. All sets are 500 randomly picked sentence subsets of the Europarl. The performance of the rule generation system is not clearly better than that of the baseline.

It is also interesting to see the number of unknown words. In Figure 7.2 are plotted the number of unknown words in the baseline system and in the system using the rule generation scheme, when translating the development set. It is obvious that the number of unknown words is much smaller in the case of generation rules. However the performance is improved accordingly terms of BLEU/NIST score. This is an indication that while the vocabulary is extended enough to include previously unknown words, the correspondence between source and target language phrase pairs has not been correctly established.

7.4 Translation table level system combination

The baseline system is interpolated with the rule generative system from Section 7.3. A good value for α has to be approximated experimentally. In order to do that translation tables for different α values were constructed and used to translate the test set. The results are shown in Figure 7.3. The best score was achieved for $\alpha = 0.2$. For this value of α Minimum Error Training was performed on the development set and translation quality was measured on the test sets. The results are in Tables 7.5 and 7.6 for the BLEU and NIST metrics respectively. It is obvious that the interpolation improves on the performance of the rule generation system a little across the different test sets. What is interesting is that the four feature variant gains

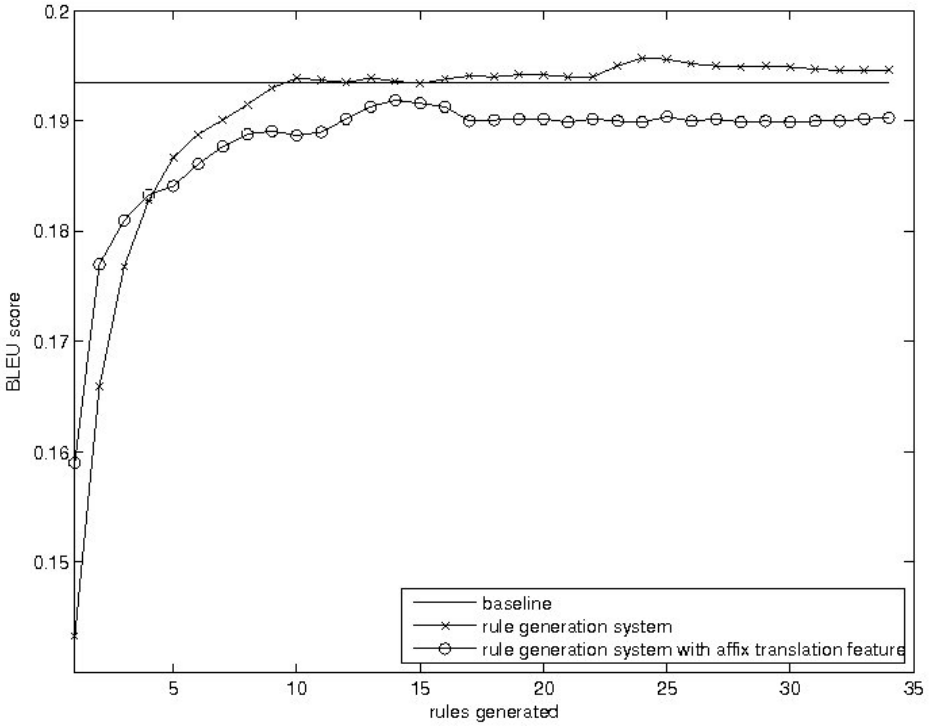

Figure 7.1: Impact of ratio of lexical rules generated per stem rule in BLEU score

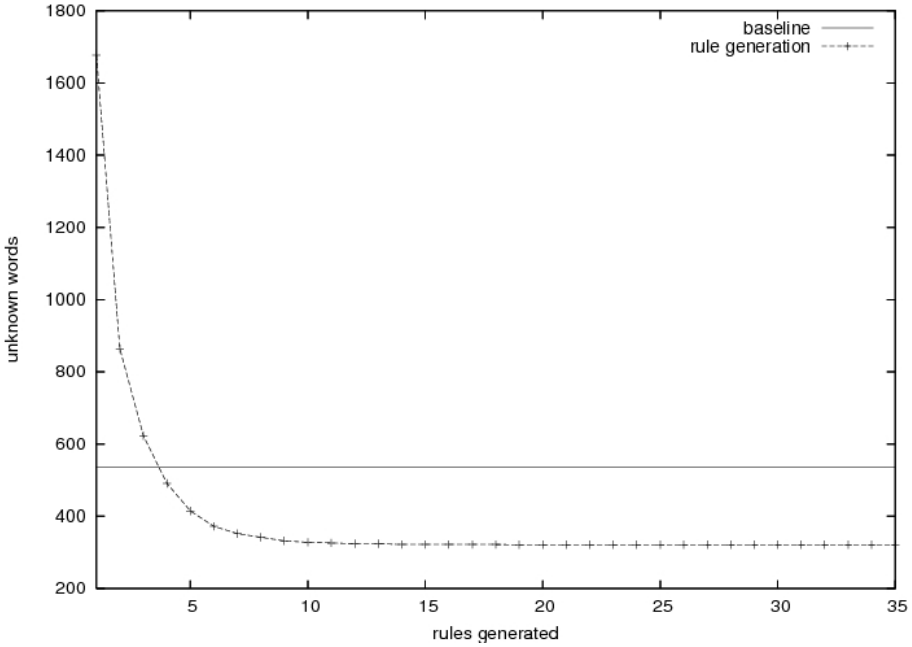

Figure 7.2: Impact of ratio of lexical rules generated per stem rule in the amount of unknown words

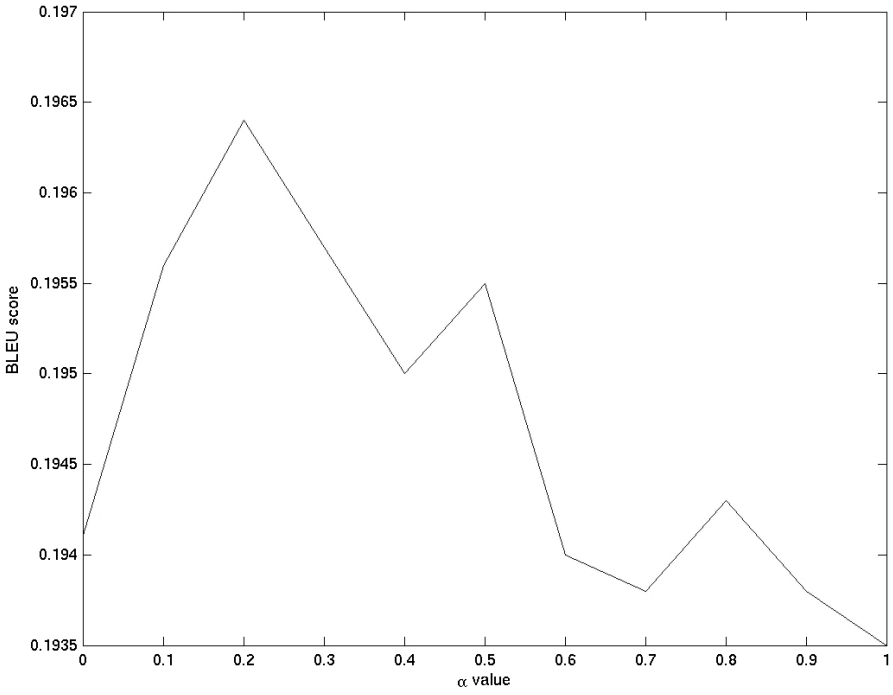

Figure 7.3: BLEU score of interpolation system as a function of α

significant improvements over the rest of the systems, but only regarding the NIST metric. This could be an indication that the further tuning is needed.

7.5 Sentence level combination

7.5.1 Sentence level combination upper limit

Independently of the approach used, it is convenient to know what is the optimum performance possible by combining the output of the systems in the sentence level. For the optimized versions of the baseline and the rule generation system, each sentence translated is scored by the TER metric [31]

[2]. Then for each sentence translated by the two systems, the highest scoring is chosen. The results are shown in Tables 7.3 and 7.4. The column *oracle max* is the translation constructed with the aforementioned process. The gain in performance is consistently at least 1 BLEU point above the baseline system. In addition to the best possible translation, the worst possible was computed, using the same process and the results are in the column denoted *oracle min*. It is interesting to note that the baseline system performance is generally halfway between the lower and the upper bound.

test set	Baseline	generative	oracle max	oracle min
dev set	0.2112	0.2138	0.2313	0.1934
1	0.2021	0.1990	0.2136	0.1868
2	0.1941	0.1944	0.2056	0.1825
3	0.2063	0.1991	0.2175	0.1869
4	0.2045	0.2040	0.2195	0.1879
5	0.2065	0.2048	0.2195	0.1898
6	0.2104	0.2133	0.2263	0.1964
7	0.2137	0.2078	0.2249	0.1959
8	0.2122	0.2119	0.2248	0.1982
9	0.2168	0.2098	0.2269	0.1978

Table 7.3: Oracle translation BLEU scores of weight optimized baseline and rule generation system combination in different test sets

7.5.2 System combination using decoder scores

The results for the combination based on decoder scores are given in Tables 7.5 and 7.6, in the column *decoder score combination*. The system outperforms the baseline in 8 out of 10 sets (including the development set). It is important to note that although tuning was done in respect to the BLEU

[2]It is preferred over BLEU as it provides more manageable figures on per sentence basis

test set	Baseline	generative	oracle max	oracle min
dev set	5.5309	5.5453	5.8049	5.2772
1	5.5044	5.4769	5.7464	5.2362
2	5.4188	5.4156	5.6787	5.1498
3	5.5137	5.4127	5.7236	5.1994
4	5.5145	5.4639	5.7283	5.2470
5	5.5247	5.4776	5.7742	5.2371
6	5.4659	5.4842	5.7336	5.2183
7	5.6019	5.5347	5.8105	5.3372
8	5.6089	5.5965	5.8436	5.3692
9	5.7033	5.5555	5.8901	5.3884

Table 7.4: Oracle translation NIST scores of weight optimized baseline and rule generation system combination in different test sets

metric only, the results are consistent across the two different metrics. This ensures that the improvement noticed corresponds to translation quality improvement and not exploitation of one metric's particular deficiency. The improvement, however, is not consistent across all experiments, since for some cases the difference is too small to be considered significant.

7.5.3 System combination based on input

In order to combine the systems output depending on the input sentence we need to find the optimum values for the two constants R_o and T. Using test set 1, we combined the systems for different values of R_o and T. The optimum value, with reference to the BLEU score of the produced translation, are $T = 3$ and $R_o = 1$. In other words, we only consider words which have appeared at most 3 times in the train corpus, and accept the baseline system translation if their mean frequency is at least equal to 1. The performance of this combination is shown in Tables 7.5 and 7.6, for the BLEU and NIST

metrics respectively. While the performance is clearly improved on the development set, in the test sets the results are less satisfactory. This means that the classification of sentences using the mean frequency of appearance of the words in the train corpus is not robust.

	System					
test set	baseline	generative	interpolation	interpolation 4 columns	decoder score combination	train frequency combination
dev set	0.2112	0.2138	0.2126	0.2121	0.2163	0.2189
1	0.2021	0.1990	0.1986	0.1986	0.2041	0.2023
2	0.1941	0.1944	0.1940	0.1923	0.1948	0.1929
3	0.2063	0.1991	0.2060	0.2009	0.2076	0.2089
4	0.2045	0.2040	0.2070	0.2059	0.2048	0.2071
5	0.2065	0.2048	0.2076	0.2057	0.2093	0.2071
6	0.2104	0.2133	0.2113	0.2120	0.2159	0.2122
7	0.2137	0.2078	0.2142	0.2102	0.2116	0.2134
8	0.2122	0.2119	0.2116	0.2083	0.2163	0.2135
9	0.2168	0.2098	0.2145	0.2166	0.2132	0.2151

Table 7.5: BLEU scores for baseline, rule generation, translation table, decoder score based and input frequency based combination systems with optimized weighting scheme

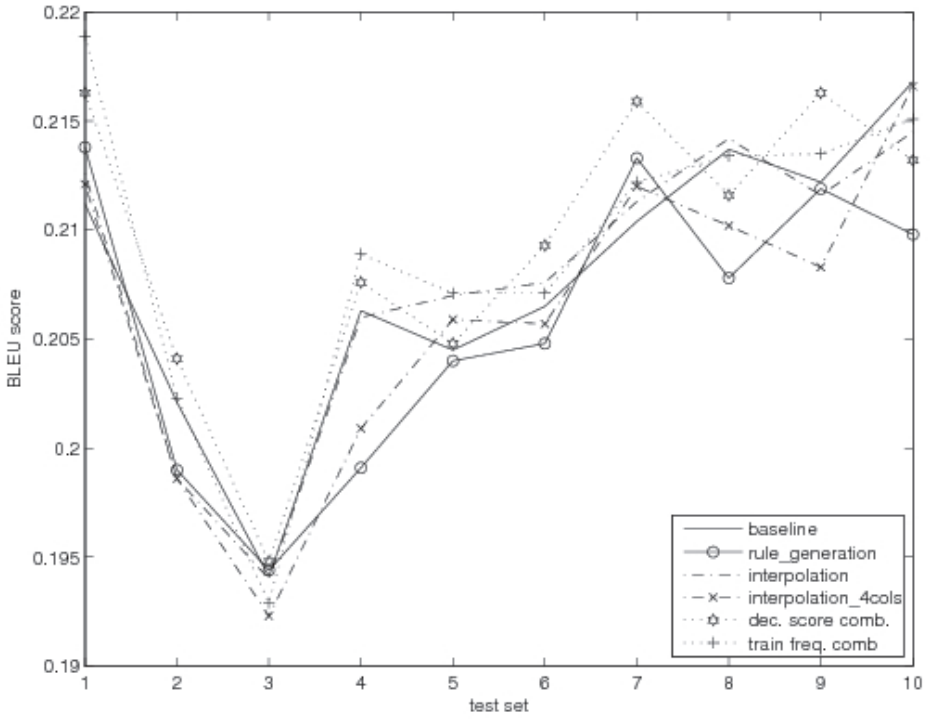

Figure 7.4: BLEU score in different test sets for baseline, rule generation, translation table, decoder score based and input frequency based combination systems with optimized weighting scheme

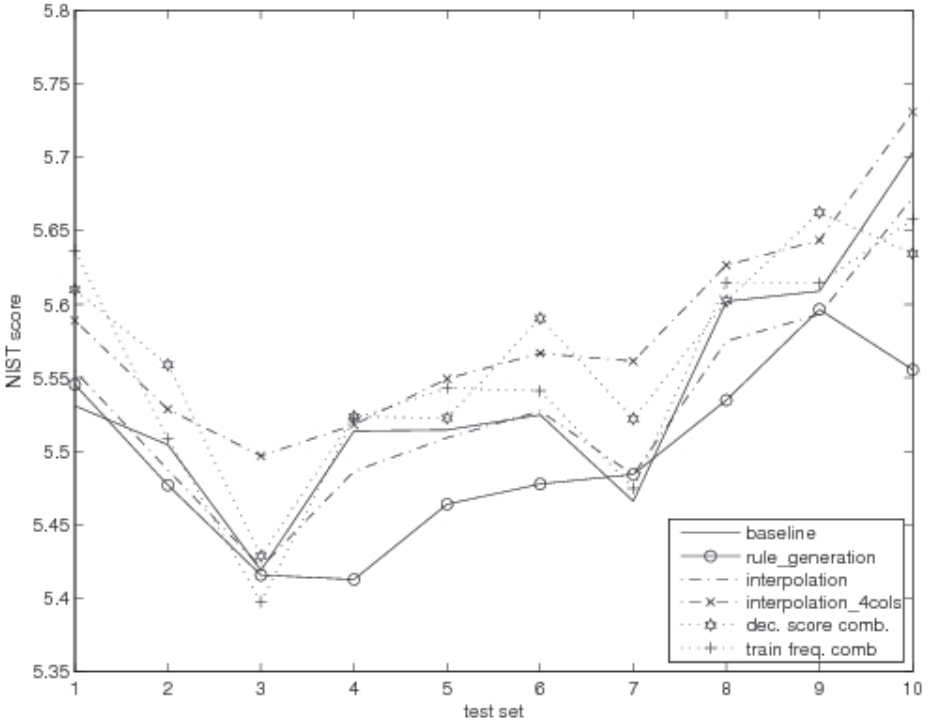

Figure 7.5: NIST score in different test sets for baseline, rule generation, translation table, decoder score based and input frequency based combination systems with optimized weighting scheme

	System					
test set	baseline	generative	interpolation	interpolation 4 columns	decoder score combination	train frequency combination
dev set	5.5309	5.5453	5.5546	5.5889	5.6099	5.6367
1	5.5044	5.4769	5.4873	5.5284	5.5588	5.5085
2	5.4188	5.4156	5.4211	5.4968	5.4286	5.3974
3	5.5137	5.4127	5.4863	5.5180	5.5236	5.5221
4	5.5145	5.4639	5.5091	5.5493	5.5224	5.5430
5	5.5247	5.4776	5.5272	5.5666	5.5904	5.5410
6	5.4659	5.4842	5.4839	5.5612	5.5220	5.4747
7	5.6019	5.5347	5.5750	5.6266	5.6020	5.6147
8	5.6089	5.5965	5.5931	5.6435	5.6625	5.6142
9	5.7033	5.5555	5.6726	5.7308	5.6345	5.6577

Table 7.6: NIST scores for baseline, rule generation, translation table, decoder score based and input frequency based combination systems with optimized weighting scheme

Chapter 8

Conclusions

In this work we have experimented with a novel approach into incorporating morphological information into a phrase based SMT system. The experimental results are encouraging. Small performance improvements have been achieved, while the amount of out of vocabulary words has dropped by more than half.

Possible ways to build up on this research line include:

1. Experimentation with the features used to compute the probabilities of the generated rules.

2. Better pruning criteria, to limit the possible generations. Instead of using a constant threshold, it could be taken into account features like the length of the phrase, the frequency of the tokens in the stem corpus, the probability of the generations that produce the resulting lexical rule.

3. Better generation model. Currently only the stem at hand is taken into account. It is almost sure that better results can be achieved by

judging also on the adjacent stems and generations.

4. It would be interesting to try the system in the inverse translation direction, from English into Greek. Translating into a morphology rich language may result in bigger improvements upon the baseline system.

5. Incorporate morphologic information into the language model. This should not be very difficult, since the language model toolkit used in this work offers the option to use factored based models.

Appendix A

Translation Examples

Below are listed some translation examples from the test sets used. The first translation is from the **baseline** system and the second one from the **rule generation** system.

παρακαλώ λοιπόν να διορθωθεί στην ημερήσια διάταξη της πέμπτης αλλά και στο εξής αυτό να ληφθεί υπόψη σε κάθε έγγραφο που διακινείται επισήμως εδώ στο χώρο μας

i would therefore be corrected in the agenda for thursday but also in that it should be taken into account in any document which διακινείται officially here in our own area

i would ask you to be on the agenda for thursday but also in the future this will be borne in mind in any official documents sent here in our area

για να επιτευχθεί αυτό πρέπει να δημιουργήσουμε ένα κανονιστικό περιβάλλον που να ευνοεί το κεφάλαιο επιχειρηματικών συμμετοχών ενώ παράλληλα χρειάζεται να θεσπισθούν ορισμένα κανονιστικά μέτρα

to achieve this we need to create a regulatory environment which is favourable to the capital summetoq'wn while we need to adopt a number of measures kanonistik'a business
to achieve this we need to create a regulatory environment which is favourable to the venture capital while we need to introduce some regulatory measures participation

κατά συνέπεια υποστηρίζω πλήρως την άποψη του κοινοβουλίου επί του σημείου που καλεί την επιτροπή να μας κρατά ενήμερους σε τακτικά διαστήματα για τις τελευταίες εξελίξεις

i therefore fully support the view of parliament on this point which calls on the commission to give us κρατά ενήμερους in regular basis of the latest developments
i therefore fully support the view of parliament on this point which calls on the commission to inform us κρατά in regular basis of the latest developments

δε κύριε πρόεδρε κύριε επίτροπε κυρίες και κύριοι η έκθεση είναι πολύ καλή όπως ακούσαμε ήδη

mr president commissioner ladies and gentlemen this report is excellent as we have already heard
mr president commissioner ladies and gentlemen this report is very good as we have already heard

η ανάπτυξη πρέπει να υποστηριχθεί καλύτερα

this development must be better

growth must be better

μέσω επενδύσεων στην ίδρυση νέων επιχειρήσεων δημιουργούνται και θέσεις απασχόλησης

through investment to the establishment of new companies and create jobs
through the establishment of new companies creating jobs and investment

σε πολλές περιπτώσεις υπάρχει και η περιβόητη ζούγκλα από κανόνες έντυπα και γραφεία που εμποδίζουν τους μικροεπιχειρηματίες να προσλάβουν προσωπικό

in many cases there is also the περιβόητη jungle of rules and offices which hinder the μικροεπιχειρηματίες to προσλάβουν staff away
in many cases there is also the notorious jungle of documents and office which prevents the μικροεπιχειρηματίες to recruit staff regulations

η προβληματική είναι γνωστή σε όλους μας έχουμε βέβαια φραγμούς που δυσχεραίνουν σημαντικά την ανάληψη κινδύνου και το επιχειρείν

the problem is known to all of us have of course barriers which δυσχεραίνουν considerably the taking risks and the επιχειρείν
the problem is known to all of us have of course which import barriers hindering the taking risks and the επιχειρείν

η όλη ιδέα δίνει όμως και την αφορμή για την άσκηση κρι-

τικής

the whole idea it but also the opportunity to exercise them
the whole idea it but also the opportunity for exercising criticism

η εμπειρία μας δείχνει το αντίθετο

experience shows quite the reverse
experience shows us the opposite

και στις δύο περιπτώσεις υπάρχει σαφής υποστήριξη της κοινοτικής στρατηγικής για την ανάπτυξη των επιχειρηματικών κεφαλαίων στην ευρωπαϊκή ένωση

in both cases there is a clear support for the community strategy for the development of business capital within the european union
and in both cases there is a clear support for the community strategy for the development of venture capital in the european union

αν αναλύσουμε τα αριθμητικά στοιχεία με απόλυτα κριτήρια θα πρέπει να είμαστε πολύ ικανοποιημένοι με αυτό το αποτέλεσμα

if we analyse the figures with completely criteria will have to be very pleased with this result
if we analyse the figures in total criteria will have to be very satisfied with this result

υπάρχουν ίσως μόνο μία ή δύο χώρες μεταξύ των όπου οι

εν λόγω επενδύσεις λειτουργούν κανονικά

there are perhaps one or two countries between where these investments are normally

perhaps there is only one or two countries between where these investments function normally

η θέση αυτή είναι μείζονος σημασίας για τις προσπάθειές μας για την παγίωση εμβάθυνση και διεύρυνση του πολιτικού διαλόγου

the position is vitally important for our efforts to παγίωση deepening and widening the political dialogue

the position that is essential for our effort to consolidating deepening and enlargement the political dialogue

η δεύτερη διάσκεψη κορυφής asem που διεξήχθη στο λονδίνο το κατέληξε στην επιτυχή έκβαση της κρίσης

the second summit asem held in london the subscribes to the successful outcome of the crisis

secondly the asem summit held in london the conclusion to the successful outcome of the crisis

το ίδρυμα ασίασευρώπης συνέβαλε σημαντικά στην επίτευξη αυτού του στόχου

the institution ασίασευρώπης contributed substantially to achieving this objective

the establishment of ασίασευρώπης made a significant contribution to achieving this objective

πρέπει να αξιοποιήσουμε την ευκαιρία αυτή που μας προσφέρει και τη δυνατότητα να εισακουστούν εκεί οι αρχές μας

we must use this opportunity and which offers us the opportunity to εισακουστούν where the authorities us
we must take advantage of this opportunity and give us the opportunity to have heard that the us authorities

η ανταλλαγές μεταξύ σχολείων πανεπιστημίων και επιχειρηματιών υπόσχονται πολλά και το ίδρυμα ευρώπησασίας κάνει καλή δουλειά

the exchanges between schools πανεπιστημίων and business for many and the institution ευρώπησασίας done a good job
the university exchanges between schools and businesses for many and the establishment of ευρώπησασίας done a good job

γιατί δεν πιάνουν το μπαλάκι η ιαπωνία ή η κίνα

why not πιάνουν the μπαλάκι the japan and china
why not πιάνουν the μπαλάκι the china or japan

νομίζω επίσης ότι η αισιοδοξία που επιδείξαμε όσον αφορά τη διαδικασία επανένωσης της κορέας ήταν λίγο υπερβολική

i also believe that the optimism that επιδείξαμε with regard to the process

of επανένωσης kor'eas was a little excessive
i also believe that the optimism that we show in the process of reunification korea was a little excessive

η κυβέρνηση του **wahid** έχει καλές προθέσεις και επιθυμεί ειλικρινά να διορθώσει τις πράξεις παλαιών καθεστώτων

the government of the wahid has good intentions and wishes to sincerely for their actions former regimes
the government of wahid has good intentions and i wish to correct deeds old regime

προφανώς η κατάσταση στη μέση ανατολή θα είναι ένα από τα θέματα τα οποία θα πρέπει να συζητηθούν στην εν λόγω διάσκεψη κορυφής παρότι είναι ανεπίσημη clearly the case in the middle east is one of the issues which will need to be discussed in this summit although it is informal
clearly the case in the middle east will be one of the issues which should be discussed in this summit although it is information

και αφού αντικειμενικά δεν μπορεί τότε μια νησιωτική χώρα που αναγκαστικά πρέπει να έχει πολλά τέτοια λιμάνια είναι δίκαιο να μην μπορεί να έχει τα ανάλογα οφέλη ως προς την λιμενική της υποδομή

and since this is a country which has must have many more such ports is fair could not have the benefits in terms of the structure of λιμενική depen-

ding νησιωτική can then
and since then an island country which inevitably must have many more such ports is fairly could not have the relevant benefits in terms of port infrastructure cannot objectives

οι λιμένες των παραμεθόριων περιοχών έχουν ιδιαίτερη σημασία παρόλο που ο όγκος του φορτίου και ο αριθμός των επιβατών είναι χαμηλότεροι από ότι στις κεντρικές περιοχές

the port of border areas are particularly important even though the volume of residues and the number of passengers is χαμηλότεροι than in the central regions
the ports of border regions are particularly important although the volume of freight and the number of passengers is χαμηλότεροι than in the central regions

αισθάνομαι ιδιαίτερη ικανοποίηση για το γεγονός ότι έχουν συμπεριληφθεί στο σύνολο οι στρατηγικές αξιολογήσεις των επιπτώσεων

i am particularly pleased about the fact that they included on all the strategies evaluations of the consequences
i am particularly pleased about the fact that they included in all the strategic impact evaluations

δεν θα τονίσω τα σημεία της έκθεσης με τα οποία συμφωνώ ωστόσο θα ήθελα να τονίσω κάποια θέματα που απασχολούν

τόσο εμένα όσο και κάποιους άλλους βουλευτές

we would stress the points in the report with which i agree but i should like to stress a few points which are both myself and some other members
i would single out the points in the report with which i agree but i should like to stress a few points concerning both myself and some other members

δεν μπορεί να επιδιώκουμε κάτι τέτοιο

we are not able to do so
we cannot do this

θαλάσσιοι λιμένες τροπολογίες και

θαλάσσιοι ports amendments nos and
maritime ports amendments nos and

Bibliography

[1] http://www.tartarus.org/~martin/PorterStemmer/.

[2] *Textpro tools demo*, http://tcc.itc.it/projects/textpro/index.php.

[3] ALPAC, *Languages and machines: computers in translation and linguistics*, Tech. report, Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, DC, 1966.

[4] Rafael E. Banchs and Haizhou Li, *Exploring spanish-morphology effects on chinese-spanish smt*, MATMT 2008: Mixing Approaches to Machine Translation (Donostia-San Sebastian [Spain]), 2008, pp. 49–53.

[5] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, *A statistical approach to machine translation*, Computational Linguistics **16** (1990), no. 2, 79–85.

[6] Stanley F. Chen and Joshua Goodman, *An empirical study of smoothing techniques for language modeling*, Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics (San Francisco) (Arivind Joshi and Martha Palmer, eds.), Morgan Kaufmann Publishers, 1996, pp. 310–318.

[7] Simon Corston-Oliver and Michael Gamon, *Normalizing german and english inflectional morphology to improve statistical word alignment*, AMTA, 2004, pp. 48–57.

[8] H. Déjean, *Morphemes as necessary concepts for structures: Discovery from untagged corpora*, 1998, University of Caen-Basse Normandie.

[9] William A. Gale and Kenneth W. Church, *A program for aligning bilingual corpora*, 1993.

[10] J. Goldsmith, *Unsupervised learning of the morphology of a natural language*, Computational Linguistics **27** (2001), no. 2, 153–198.

[11] Sharon Goldwater and David McClosky, *Improving statistical MT through morphological analysis*, HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (Vancouver, British Columbia, Canada), Association for Computational Linguistics, 2005, pp. 676–683.

[12] Jonathan Graehl, *Carmel finite-state toolkit*, 1997, http://www.isi.edu/licensed-sw/carmel/.

[13] Zellig Harris, *Methods in structural linguistics*, The University of Chicago Press, 1951.

[14] Ioannis Klasinas, *Statistical machine translation incorporating morphological knowledge and using improved alignments*, Diploma thesis (Technical university of Crete, Chania), 2006.

[15] Philipp Koehn, *Pharaoh: A beam search decoder for phrase-based statistical machine translation models*, AMTA, 2004, pp. 115–124.

[16] ———, *Europarl: A parallel corpus for statistical machine translation*, MT Summit, 2005.

[17] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst, *Moses: Open source toolkit for statistical machine translation*, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session (Prague, Czech Republic), June 2007.

[18] Young-Suk Lee, *Morphological analysis for statistical machine translation*, HLT-NAACL 2004: Short Papers (Boston, Massachusetts, USA) (Daniel Marcu Susan Dumais and Salim Roukos, eds.), Association for Computational Linguistics, May 2 - May 7 2004, pp. 57–60.

[19] Wolfgang Macherey and Franz J. Och, *An empirical study on computing consensus translations from multiple machine translation systems*, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 986–995.

[20] Einat Minkov, Kristina Toutanova, and Hisami Suzuki, *Generating complex morphology for machine translation*, Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (Prague,

Czech Republic), Association for Computational Linguistics, June 2007, pp. 128–135.

[21] Sonja Nießen and Hermann Ney, *Statistical machine translation with scarce resources using morpho-syntactic information*, Computational Linguistics **30** (2004), no. 2, 181–204.

[22] F. J. Och, *Minimum error rate training in statistical machine translation*, In Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL) (Sapporo, Japan), July 2003, pp. 160–167.

[23] F. J. Och and H. Ney, *A systematic comparison of various statistical alignment models*, Computational Linguistics **29** (2003), no. 1, 19–51.

[24] National Institute of Standards and Technology, *Automatic evaluation of machine translation quality using n-gram co-occurrence statistics*.

[25] Giorgos Orphanos and Christos Tsalidis, *Combining handcrafted and corpus-acquired lexical knowledge into a morphosyntactic tagger*.

[26] Kishore Papineni, Salim Roukos, and Todd Wardand Wei-Jihg Zhu, *Bleu: a method for automatic evaluation of machine translation*, In proceedings of ACL, 2002.

[27] Daniel Marcu Philipp Koehn, Franz Josef Och, *Statistical phrase-based translation*, In Proceedings of 2003 HLT/NAACL, 2003.

[28] Maja Popović and Hermann Ney, *Towards the use of word stems and suffixes for statistical machine translation*, International Conference on

Language Resources and Evaluation (Lisbon, Portugal), may 2004, pp. 1585–1588.

[29] Jorma Rissanen, *Stochastic complexity in statistical inquiry theory*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1989.

[30] Φανούρης Μωραΐτης , *Συστήματα Αυτόματης Μετάφρασης Χρησιμοποιώντας Στατιστικά Μοντέλα* , Diploma thesis (Technical University of Crete, Electronic & Computer Engineering, Chania), 2004.

[31] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul, *A study of translation edit rate with targeted human annotation*, Proceedings of Association for Machine Translation in the Americas, 2006.

[32] A. Stolcke, *SRILM – an extensible language modeling toolkit*, In Proc. Intl. Conf. on Spoken Language Processing, 2002.

[33] Jinxi Xu and W. Bruce Croft, *Corpus-based stemming using cooccurrence of word variants*, ACM Transactions on Information Systems **16** (1998), no. 1, 61–81.

[34] Mei Yang and Katrin Kirchhoff, *Phrase-based backoff models for machine translation of highly inflected languages*, EACL, 2006.

[35] Andreas Zollmann, Ashish Venugopal, and Stephan Vogel, *Bridging the inflection morphology gap for arabic statistical machine translation*, Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers (New York City, USA), Association for Computational Linguistics, June 2006, pp. 201–204.