TOWARDS INCORPORATING MORPHOLOGY INTO STATISTICAL MACHINE TRANSLATION

By

Panagiotis D. Karageorgakis

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
TECHNICAL UNIVERSITY OF CRETE
CHANIA, GREECE
NOVEMBER 2005

TECHNICAL UNIVERSITY OF CRETE
DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Towards Incorporating Morphology into Statistical Machine Translation**" by **Panagiotis D. Karageorgakis** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: November 2005

Supervisor: _____
Assoc. Prof. Alexandros Potamianos

Readers: _____
Prof. Vasilis Digalakis

_____
Prof. Stavros Christodoulakis

TECHNICAL UNIVERSITY OF CRETE

Date: **November 2005**

Author: **Panagiotis D. Karageorgakis**

Title: **Towards Incorporating Morphology into Statistical Machine Translation**

Department: **Electronics and Computer Engineering**

Degree: **M.Sc.** Convocation: **November** Year: **2005**

Signature of Author

To my family and to Fenia

Table of Contents

Table of Contents v

List of Figures viii

Abstract ix

Acknowledgements x

Introduction 1

1 Background in Statistical Machine Translation 5
- 1.1 Introduction to Machine Translation . . . 5
  - 1.1.1 What is Machine Translation? . . . 5
  - 1.1.2 A Brief History of MT . . . 6
  - 1.1.3 Approaches to MT . . . 9
- 1.2 Statistical Machine Translation . . . 12
  - 1.2.1 Overview . . . 12
  - 1.2.2 Bayes Rule and the noisy channel model . . . 13
  - 1.2.3 Faithfulness and fluency in translation . . . 14
  - 1.2.4 Computing models . . . 16
  - 1.2.5 Decoding . . . 17

2 Language Morphologies 19
- 2.1 Introduction . . . 19
  - 2.1.1 What is a language's morphology? . . . 19
  - 2.1.2 Different kinds of morphologies . . . 19
- 2.2 Learning a morphology . . . 21
- 2.3 Related work into morphology and SMT . . . 21
  - 2.3.1 Bootstrapping using a knowledge source . . . 22
  - 2.3.2 Obtaining affix inventories . . . 25
  - 2.3.3 Performing a complete morphological analysis . . . 27
- 2.4 The Linguistica Morphological Analyzer . . . 32

2.4.1 Improving Linguistica 33

3 The Baseline SMT System **36**
3.1 Overview 36
3.2 Sentence Boundary Detection 37
3.3 Sentence alignment 38
3.4 Language modeling 39
3.5 Translation modeling 40
3.5.1 Word-based translation modeling 40
3.5.2 Phrase-based translation modeling 41
3.6 Decoding 42

4 Morphology Incorporation into SMT **44**
4.1 Overview 44
4.2 Motivation 46
4.3 Addressing data sparseness problems 47
4.4 System Architecture 47
4.4.1 Morphological Analyzer 48
4.4.2 Stem-level SMT system 49
4.4.3 Morphological generator 50
4.5 Statistical framework 51
4.5.1 Lexical SMT system 51
4.5.2 Stem to stem translation 51
4.5.3 Word-to-word translation via the stem-to-stem system . . 52
4.5.4 SMT system combination 53
4.6 Combined Lexical-Morphological system implementation 54
4.7 Incorporating affix information 57

5 Experimental Results **58**
5.1 Overview 58
5.2 Evaluation Metrics 59
5.2.1 The BLEU metric 60
5.2.2 The NIST metric 60
5.3 Morphological analysis 61
5.4 Corpora 62
5.5 Parameters 62
5.5.1 Parameter 1: Training corpus size 63
5.5.2 Parameter 2: Combination weights 64
5.6 Results 65
5.7 Statistical Significance 70
5.7.1 Practical limitations 70

5.7.2 Confidence intervals 70
5.7.3 Bootstrap resampling 71
5.7.4 Statistical significance test results 72

6 Conclusions **74**
6.1 Ongoing work 76
6.1.1 Affix-to-affix translation 76
6.1.2 Extended SMT system 76
6.1.3 Additional affix information incorporation 78
6.2 Future work 81

A Translation samples **83**

Bibliography **87**

List of Figures

4.1 System Architecture 49

5.1 Improvement in NIST scores (over w_1) 66
5.2 Improvement in BLEU scores (over w_1) 67
5.3 Improvement in NIST scores (over training set) 68
5.4 Improvement in BLEU scores (over training set) 69

6.1 Enhanced Bayesian Network Topology 77

Abstract

In this work, a novel algorithm for incorporating morphological knowledge into statistical machine translation (SMT) systems is proposed. Using an unsupervised algorithm, morphological analysis is performed for the texts of the source and target languages resulting in rules that specify the separation of each word into its stem and affixes. Stem information is used for the construction of a stem-level SMT system, that translates from stems of one language to stems of another. Using a general statistical framework that we are proposing, the stem-level SMT system is combined with a lexical SMT system, resulting in word-to-word translation through the morphological SMT system.

The combined lexical-morphological SMT system has been implemented using late integration and lattice re-scoring. The output of this system is compared to that of the baseline lexical system, after training both of them on the Europarl corpus, a large multilingual parallel corpus. Various experiments have been conducted by altering the training set size as well as the weights that are used in the combination of the two systems. For the evaluation of the two systems translations, two automatic evaluation metrics have been used.

Experiments have shown that the scores for both the two metrics, NIST and BLEU, for the combined lexical-morphological SMT system, improve over the baseline system by 14% and 33% for the NIST and BLEU metrics respectively, for translation of English to Greek, using a 800k word training corpus, on the sentences that result in different translations between the two systems. To provide further evidence of the validity of our results, we have performed statistical significance tests to calculate confidence intervals for the true values of the improvement our system provides.

We are also proposing an enhancement to the combined lexical-morphological SMT system, by providing the theoretical framework for the incorporation of affix information into the system. Such an incorporation could take place by creating an affix-to-affix system and combining this system into the statistical framework of the combined lexical-morphological system.

Acknowledgements

I would like to thank my supervisor, Alexandros Potamianos, for his invaluable guidance and support as well as for the excellent cooperation we had during this research. I would also like to thank professor Vasilios Digalakis for his useful comments and guidance, as well as professor Stavros Christodoulakis for our cooperation while I was working at the M.U.S.I.C. Laboratory of the Department.

Many thanks go to Fanouris Moraitis for developing the baseline system that have been used in this research as well as to Ioannis Klasinas for providing enhancements to it. I feel obliged to thank Chris Vosnidis, not only for his good friendship but also for offering a helping hand whenever needed, thus softening the process of entering a territory of science I was rather unfamiliar with. Also, for lending me books (for research purposes) and computer gear (for fun and work).

My dearest thanks go to my family (my parents and my brother, George) for their continuous support, love and understanding throughout my whole life and during this research as well. It would be unfair not to mention that the driving force behind this research has been the person who brought me to the island of Crete in the first place, therefore I would like to express my dearest thanks to Fenia, for standing always by my side and brightening things up.

Last, but not least, I wish to thank my friends, Apostolis Pangos (also for walking the same path with me since the first day in Chania), Michalis Toutoudakis and George Ligoksygakis for making everyday life easier because of their company, friendship and helping hand in my darkest hours during these three years.

Chania, Crete — Panagiotis Karageorgakis
November 4, 2005

Introduction

The translation of foreign language texts by computers has been the driving force for the research that has been carried out during the past 50 years in the field of Machine Translation (MT). A matter of speculation long before computers were capable of performing complex calculations to make such a tedious task viable, the field of MT has claimed attention from pioneers in linguistics, philosophy, computer science and mathematics.

Today MT has become an important field of research and development as the need for translation of technical and commercial documentation is growing. Great progress has been made, yet the translation quality of today's state-of-the-art systems is nowhere near that of a skilled human professional translator. This is mainly due to the inherent complexities that characterize human languages and the inability of current systems to exploit the wealth of information within them, such as linguistic information.

Different approaches to Machine Translation (MT) exist today. *Statistical Machine Translation* (SMT) is the approach that utilizes the power of statistics, by training statistical models for sets of languages and using Bayesian inference to compute the best translation from one language to the other. Most existing SMT systems operate at the lexical level, providing word-based or phrase-based translation by modeling word or phrase relations between sets of languages. Such systems do not take advantage of the linguistic information that is inherent in

the training sets used, such as the morphology of each language.

Morphology is the study of the way words are built up from smaller meaning bearing units, such as stems and affixes. An analysis of large monolingual corpora may provide rules that specify how each word of the vocabulary of the language is constructed by a stem and optional affixes in an unsupervised way. Algorithms for such a morphological analysis exist today, one of which is used in our work for the derivation of such rules.

In this work, we are proposing a method for the incorporation of linguistic knowledge, specifically morphological knowledge, into existing SMT systems. First, morphological rules are extracted in in unsupervised way, resulting into knowledge about the way each word can be split into a stem and its affixes. This knowledge is used in order to stem the training corpora and to train statistical models that allow for the creation of a system that performs translation at the stem level. The stem-level SMT system is then combined with the lexical SMT system, resulting on what may be called a combined lexical-morphological SMT system. This combination employs weights that model the participation of each system in the process of translation, allowing it to give more weight to one of the two systems.

In this work, we are also demonstrating an implementation of such a lexical-morphological SMT system, using late integration and lattice re-scoring. During the stage of decoding, which is the process of computing the best translation of a given sentence, lattices are produced for both the word-level and stem-level system. These lattices are represented as weighted finite state machines, as also happens for the stem-to-word model which implements what we call a morphological generator. The combination of the two systems is then implemented by operations between the resulting finite state machines.

Evaluation of both systems output is done automatically, in order for the results to be unbiased, using two automatic evaluation metrics that are widely used in Machine Translation, namely the BLEU and NIST metrics. These metrics are computed by comparing the translation of a system to a reference translation, that has been created by a human expert. Computing scores for these metrics makes it possible to directly compare the translation quality of the two different SMT systems. This comparison has shown that the lexical-morphological SMT system greatly improves the translation quality, up to 33% and 14% for the two metrics respectively, for a 800k words training corpus, on the sentences that result in different translations between the two systems.

The evaluation of any SMT system is affected by the size of the testing sets. Although we used quite large testing sets, in order to be confident that the results are valid we performed statistical significance tests. Using the method of bootstrap resampling, we calculated confidence intervals for the true value of the improvement achieved by the morphological knowledge incorporation. These tests have shown that our results are statistically significant and that there is clear improvement in the translation quality.

Document Structure

The rest of this document is structured as follows: Chapter 1 provides the essential background in the field of Statistical Machine Translation (SMT), while Chapter 2 discusses language morphologies as well as related work in the field of SMT research. The baseline system that has been used is introduced in detail in Chapter 3, while the theoretical framework of our work as well as an implementation of such a system are presented in Chapter 4. Chapter 5 discusses the

experiments that have been conducted and demonstrates the results of the comparison between the new system and the baseline. We conclude our research in Chapter 6 and provide some pointers for future work in this area. Some translation examples from both systems are presented in Appendix A.

Chapter 1

Background in Statistical Machine Translation

1.1 Introduction to Machine Translation

1.1.1 What is Machine Translation?

Before discussing about *Statistical* Machine Translation, it is useful to provide some brief introduction to the field of Machine Translation in general. The European Association for Machine Translation gives the following definition for MT: "*Machine translation (MT) is the application of computers to the task of translating texts from one natural language to another*" [11]. The term *Machine Translation (MT)* is the traditional and standard name for computerised systems responsible for the production of translations from one natural language into another, with or without human assistance. Other terms have been used in the past, such as "mechanical translation" and "automatic translation" but are rarely used now in English.

1.1.2 A Brief History of MT

The idea of using mechanical dictionaries to overcome the limitations of languages is not new; both Descartes and Leibniz, back in the 17th century, speculated on the creation of dictionaries based on universal numerical codes. The "universal language" movement was born, which is the idea of creating an unambiguous language, based on logical principles and iconic symbols, with which all humanity could communicate without fear of misunderstanding. The most familiar work in this area is the *interlingua*, elaborated by John Wilkins in [16].

In this work, Wilkins defines the "real character" which is a new orthography for the English language that resembles shorthand, and the "philosophical language" which is based on an early classification scheme, or *ontology*, in what would later become the computer science meaning of the term. He then describes a large number of possible concepts as single words by first dividing all reality into forty different categories, each assigned to a different syllable, then sub-dividing these categories into sub-categories, and so on. The resulting words thus encode some of the semantics of their meanings into their spelling. Such a-priori languages were inspired by accounts of how the Chinese writing system worked.

Although there were many proposals for international languages in the subsequent centuries[1], few attempts to mechanize translation took place, until the middle of the 19th century. Then, in 1933, two patents appeared independently of each other in France and in Russia; one of a French-Armenian named George Artsrouni, and the second by the Russian Petr Smirnov-Troyanskii.

[1]More modern a-priori languages are *Solresol*, an artificial language devised by Jean Franhois Sudre beginning in 1817, and *Ro*, a constructed language created by Rev. Edward Powell Foster beginning in 1904, yet the most widely spoken constructed international language is *Esperanto*, first published in 1887 by L. L. Zamenhof under the pseudonym D-ro Esperanto (Dr. Hopeful).

The latter work was indeed pioneer in the field of MT; Troyanskii's proposal consisted of three stages. First, an editor knowing only the source language was to perform a logical analysis of words into their base forms and syntactic functions. Then, another machine would transform sequences of base forms and functions into equivalent sequences in the target language. Finally, a third editor, which held knowledge of the target language only, would convert this output into the normal forms of that language[2]. Apparently, this is a rough idea of how modern MT systems work, and it is certain that the reader may find many similarities between the Russian's work back in 1933, and the architecture of the systems we are describing in later chapters of this document.

Although Troyanskii's work is quite astonishing for his era, few references about his work exist in the literature. When it comes to mentioning the people who first had the idea of translating automatically between languages, there is some dispute about conversations between Andrew D. Booth, a british crystallographer, and Warren Weaver of the Rockefeller Foundation, and more specifically to a memorandum written by the latter which included the folllowing two sentences:

> "*I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.*"

Later in the memorandum, Weaver proposed some other more sophisticated methods, that could turn such an apparent difficult task into one that could be approached with the emergent computer technology of that time. When the idea

[2]Troyanskii's patent referred only to the machine involved in the second stage, although he believed that "the process of logical analysis could itself be mechanized".

was brought to general notice, various methods were suggested, being inspired by the use of war-time cryptography techniques, statistical analysis, Shannon's information theory and exploration of the underlying logic and universal features of spoken languages. By the early 1950s there was a large number of research groups working in Europe and the USA, but despite some success, there was doubt about the possibility of automating translation in general.

The mistakes of the early MT workers have been accredited to the lack of judgement: the complexity of the conceptual problem of natural language understanding was underestimated [30]. In 1964 the National Academy of Sciences of the United States published the report of its Automatic Language Processing Advisory Committee (ALPAC), in which it was recommended that most research into MT should be stopped immediately due to its failure to produce useful translation, destroying all confidence in the vision of building a fully automated MT system.

During the next decade, researchers moved away from MT and concentrated on understanding language processing, but in the early 1980s there was a revival of MT research, though much of the work was carried out in Japan and not Europe or the US. In Europe the Commission of the European Communities (CEC) did invest in the English-French version of the SYSTRAN and in 1997 Altavista adopted BabelFish Translation making it the first real time Systran translation to appear on the Web.

The reader that is further interested into the history of MT shall find detailed information about the subject in [15].

1.1.3 Approaches to MT

As research in the field of MT has been carried out in parallel and in many research groups and countries, there have been different approaches to the problem of machine translation. The most important ones, are:

- direct translation
- transfer based translation
- interlingua
- example-based translation
- statistical translation

Direct Translation

The "direct approach" is the earliest historically, adopted by most MT systems of what has come to be known as the first generation of MT systems. It lacks any kinds of intermediate stages in the translation processes: the processing of the source language input text leads "directly" to the desired target language output text. Direct translation systems can be considered to be word-for-word systems because they do not offer something in the field of semantic, contextual knowledge etc. They are missing any analysis of the internal structure of the source text, particularly the grammatical relationships between the principal parts of the sentences.

Transfer Based Translation

The lack of contextual accuracy and the inability to capture the meaning of the source text using a direct strategy prompted the development of systems

that could. Such a strategy for translating from one language to another, is to treat translation as a process of altering the structure and words of an input sentence to arrive at a valid sentence of the target language. Application of this metaphor can be applied using knowledge about the differences between a set of two distinct languages. Systems that use this strategy are often said to be based on the transfer model.

The transfer model consists of three phases: analysis, transfer and generation. The first phase involves some sort of parsing, in order to construct a representation of the structure of the input, which in phase two is transformed into a structure for the target language. Finally, this structure is given as input to a generator, in order to actually create the output sentence.

Based on the transfer model and depending on the level of the contrastive knowledge that is modeled, such systems may utilize lexical, syntactic or semantic transformations between two languages. Lexical transfer is the process of finding the correct word in a cross-language dictionary, while syntactic transfer is the operation of mapping from one parse tree (which models a sentence), to another. At a higher level, this could also be applied to semantics, by generating semantic trees that should then be transformed into the corresponding structures of the target language, which is a rather tedious task. All of the rules that the transfer model uses, in order to perform such transformations are usually hand-written, which makes it rather expensive to use.

Interlingua

Methods based on the transfer model pose a serious disadvantage, in that they require a distinct set of transfer rules for each pair of languages. When it comes to systems that target multilingual environments, like the European Union where

there are eleven official languages, it appears that the transfer model is quite suboptimal. An alternative is to extract the meaning of the input sentence and express it using the target language. This way, there is no need for transformation rules between pairs of languages; the amount of knowledge needed is proportional to the number of languages that the system needs to handle, rather than the square as in the transfer model.

The common meaning representation that can be derived by all languages is termed *interlingua* and instantly brings to mind the works of Wilkins and the constructed languages that have been mentioned in Section 1.1.2, yet it does not have to be a real language, but a language-independent canonical form. Translation using this method first performs a semantic analysis on the input sentence and constructs the interlingual representation, then generates from this intermediate structure the sentence in the target language.

This approach however is not free from limitations. In order for such a system to work, there has to be a global interlingua vocabulary in which every word, or meaning, from every language has to exist, which does not hold true, especially for languages that have been derived from different cultures. Although the goal of defining a universal interlingua is both intellectually stimulating and has many potential advantages, it seems highly unlikely that there will ever be one truly universal interlingua.

Example-based translation

Example-based translation is essentially translation by analogy. An Example-Based Machine Translation (EBMT) system is given a set of sentences in the source language (from which one is translating) and their corresponding translations in the target language, and uses those examples to translate other, similar

source-language sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct again. Which examples an EBMT system determines to be equivalent (or at least similar enough) to the text to be translated varies according to the approach taken by the system.

Statistical Machine Translation

SMT may be viewed as a different approach to machine translation, when it is compared to the other four approaches that have been described so far, because it focuses on the result, and not on the process of translation. The goal of an SMT system is to find the most probable sentence in the target language, given a sentence in the source language. These probabilities are determined automatically by training statistical models using large corpora of bilingual texts, translated usually by human specialists. This approach in presented in detail in the next Section.

1.2 Statistical Machine Translation

1.2.1 Overview

Statistical Machine Translation is the approach to machine translation that uses probabilistic models in order to calculate the best translation of a given sentence. Section 1.2.3 discusses what is meant by the term "best" translation and the compromises that must be made in order to achieve it.

As a research area, SMT started in the late 1980s with the Candide project at IBM [4]. This original approach calculated mappings between words and allowed for deletion and insertions of words as well. Translation quality has been

improved with the use of phrase translation, as early as Och's [32] alignment template model.

1.2.2 Bayes Rule and the noisy channel model

The most fundamental role in SMT theory (as well as in other natural language applications) is the notion of Bayesian inference, or noisy channel model. This model has been based on speech recognition techniques, where the source signal has been altered by passing through a noisy communications channel, resulting into a noisy word. This word should then be decoded in order to recover the original word.

Bayes theorem is essentially an expression of conditional probabilities. Conditional probabilities represent the probability of an event occurring given evidence. The theorem can be derived from the joint probability of A and B as follows:

$$P(A,B) = P(B,A)$$

$$P(A|B)P(B) = P(B|A)P(A)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1.2.1}$$

where $P(A|B)$ is referred to as the posterior probability; $P(B|A)$ is known as the likelihood, $P(A)$ is the prior probability and $P(B)$ is generally the evidence and is used as a scaling factor. Therefore, Bayes rule can be formulated as:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Application to MT

The noisy channel model can be used in MT: the intuition is to treat the sentence of a foreign language that needs to be translated in the native language, as an original sentence in the native language that has been passed through a noisy communication channel. The sentence is transformed while passing through this channel because of the "noise"[3] of the channel. The goal then is to remove the "noise" and recover the original sentence.

Given a sentence f in a foreign language, the goal is to find a sentence n in the native language that maximizes the probability $P(n|f)$, as

$$\hat{n} = \arg\max_{n} P(n|f)$$

which by using Eq. (1.2.1) may be rewritten as:

$$\hat{n} = \arg\max_{n} \frac{P(f|n)P(n)}{P(f)}$$

and since the maximization is over n it can be finally written as:

$$\hat{n} = \arg\max_{n} P(f|n)P(n) \tag{1.2.2}$$

So in order to calculate the best translation the probabilites $p(f|n)$ and $p(n)$ should be modeled, where $p(f|n)$ models the likelihood, i.e. how likely it is to have the sentence f, given the sentence n, while $p(n)$ models the prior probability of sentence n to appear in the training corpora.

1.2.3 Faithfulness and fluency in translation

The very idea of translating from one language to another may be viewed firstly as a subjective process and secondly as incapable of producing a perfect translation. The first point is based on the ambiguities that are present in all natural

[3]Although the term "noise" has been used literally in speech recognition tasks, it is used as a metaphor in SMT.

languages, that may lead in multiple "correct" translations. However, a sentence that one person considers to be the best, may sound inappropriate for another person, making the evaluation of the outcome a subjective process. The second point is justified by the absence of 1:1 mappings between concepts and ideas between sets of languages. For example, the word *tread* may not appear in the language of a race that has not yet invented rubber wheels thus rendering it impossible to translate the sentence *the tread wears* into that race's dialect[4].

In order to overcome these limitations, the quest for the calculation of the "best" translation needs to take under consideration two different aspects: faithfulness and fluency.

Faithfulness

When multiple translations exists for a given sentence, there should be a means of measuring how close each one of them is compared to the original sentence. This is quoted as the faithfulness of a sentence. For example, the sentence *the tread wears* could be translated in two ways: one that denotes that the outer area of a tyre is being deconstructed into pieces, and one that claims that the tyre is wearing something (perhaps a cloth or accessory). Obviously, the first translation is a much more faithful attempt than the second one.

Fluency

Fluency may be defined as "the ability to express oneself readily and effortlessly". In the case of MT, fluency characterizes a sentence regarding it's validity in the language it belongs to. For example, the sentence "that is I" consists of valid

[4]In the case of this example it would be difficult in general to translate sentences in the domain of automobiles, since it is not very probable for a race that hasn't invented rubber wheels to have invented cars.

english words, yet is not valid as a sentence, in contrast to the sentence "that is me". Hence, the model $p(n)$ that models the prior probability of a sentence n to exist in the language, models the fluency of each sentence.

Going back to Eq. (1.2.2) it can be seen that the model $p(f|n)$ actually models the faithfulness of the translation. Substituting $p(n)$ and $p(f|n)$ with the corresponding concepts that they model into Eq. (1.2.2), we can express it in non-mathematical terms as:

$$\text{best-translation}\hat{T} = \arg\max_{T} \text{faithfulness} \times \text{fluency}$$

which shows that in SMT, the goal of translation may be viewed as the production of an output that maximizes some value function that represents the importance of both faithfulness and fluency. Among all the sentences that correspond to the original sentence, the one which maximizes both it's faithfulness and it's fluency is considered to be the best translation.

1.2.4 Computing models

Having defined what is considered to be the best translation of a sentence, in order for it to be calculated, the models that have been mentioned in the previous section must be computed. The model $p(n)$ is usually referred to as the *language model*, since it models the language; the model $p(f|n)$ is then referred to as the *translation model* since it models the translation of one sentence into another. Both these models can be computed by assessing large bilingual (or multilingual) parallel corpora. These corpora that are used in training the models, must be a large set of texts that are written in the native language, and the corresponding texts in the foreign language, usually translated by human experts.

Regarding the language model, it can be easily computed as an n-gram model

of the language. In order to compute the translation model though, some pre-processing needs to take place. First of all, the corresponding documents from every language must be paired. Then, the sentences that they contain must be matched, and the words or phrases aligned. Pairs of words, or phrases (in word-based or phrase-based translation accordingly) that appear in the same position in texts that represent translations of the same source are then computed and are assigned a probability, based on the number of times that they appear in the corpora. The most well known models in this area are IBM Models 1 through 5, which approximate the translation of n to f as a word substitution/permutation process. Each of the models have a slightly different generative probabilistic story. In depth information about these models may be found at [5].

1.2.5 Decoding

After the models have been computed, the best translation can be calculated. Because the set of possible translations is enormous, efficient methods must be used while searching, without actually generating the infinite set of all possible translations. Because machine translation allows for word re-ordering, finding the output that maximizes the objective function is NP-complete [22].

This decoding problem is usually addressed by a dynamic programming-based beam decoder, where the output is produced left-to-right, by incrementally constructing a lattice of partial translation hypotheses. Each one of them stores: the last pair of words/phrases used in translation, the next-to-last target word, a coverage vector that makes explicit which source words have been already translated, a language and translation model score as well as other model scores. The most promising hypotheses are expanded left to right until a translation with a coverage vector that covers the entire sentence is found, then backtracking to

generate the sentence. Instead of storing just the best translation, it is often useful to store a list of n-best translations, as well as the lattice that has been generated in the process of decoding.

Chapter 2

Language Morphologies

2.1 Introduction

2.1.1 What is a language's morphology?

Morphology is the study of the way words are built up from smaller meaning-bearing units, which are called *morphemes*. In the information retrieval domain, the similar (but not identical) problem is called *stemming*, which usually deals with removing endings from words leaving the stem (root) of the word. However, a full morphological analysis is more than that, and is usually regarded as a segmentation of the word into morphemes combined with an analysis of the interaction of these morphemes that determine the syntactic class of the word form as a whole.

2.1.2 Different 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, infix or circumfix, whether it

precedes the stem, follows it, do both or being inserted in it respectively. The use of prefixes and suffixes (and circumfixes as well, since they may be viewed as a combination of a prefix and a suffix) is 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 *non-concatenative 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** and **derivational** morphology. These two are partially overlapping since the borders between them are usually not absolutely clear. Inflection mostly deals with the use 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.

In order to build morphological parses, there are three general classes of linguistic knowledge that are needed: a *lexicon*, which is the list of stems and affixes, together with basic information about them; *morphotactics*, which is the model of morpheme ordering that explains which classes of morphemes can follow other classes of morphemes inside a word; finally *orthographic rules*, which are the spelling changes that can occur due to morpheme attachment. However, unsupervised algorithms for performing a morphological analysis are available, using statistical means.

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 linguists 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 a corpus

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

In this chapter we are discussing algorithms for the unsupervised learning of morphologies, since such methods may be used with untagged corpora which is often the case, performing a 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.

2.3 Related work into morphology and SMT

In this section, the most important approaches of (mostly) unsupervised morphology learning algorithms 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 [35]. 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

2.3.1 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.

Exploiting co-occurence

Although their work may be more suited to information retrieval (IR), Xu and Croft [37] 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.

Using inflectional lexicons

Gaussier [12] proposes an unsupervised method of learning derivational morphologies from an inflectional lexicon of a language. Apart from restricting his work

on suffixes and suffixation operators (and not prefixes, infixes or circumfixes), he further focuses only on concatenative languages where derivation can be viewed as a concatenation operation. For a given language, his method builds relational families, which are an approximation of derivational families, which are then used to produce pairs of words which are a first approximation of the pairs of related words. From this set some parameters are being estimated and then used to induce a derivation tree on each relational family. These trees are then used to refine the previous set of word pairs and suffixes and extract dependencies between suffixation operations as well as morphographemic rules. The lexicon that is being used includes part of speech information.

Multimodal alignment

Yarowsky and Wicentowsky [40] propose a corpus-based algorithm for the nearly unsupervised induction of inflectional morphological analyzers, with a focus on highly irregular forms. They obtain outstanding results at inducing English past tense (92.2% accuracy for their set). Their method requires a table of the inflectional parts of speech of the given language along with a list of the canonical suffixes for each part of speech, a list of the open class roots in the language and a large unannotated text corpus. Their method treats morphological analysis predominantly as an alignment task, performing the effective collaboration of four similarity measures: expected frequency distributions, context, morphologically-weighted Levenshtein similarity and a model of affixation and stem-change probabilities. This combination of different probabilistic models finds pairs of words that are likely to be morphologically related.

Learning of morphology without morphemes

An interesting approach against the literature of computational morphology, is that of Neuvel and Fulop [29], who propose a mechanism for the induction of morphological relationships without attempting to discover or identify morphemes, using a part-of-speech tagged lexicon as an input. Additionally, their method claims to be able to generate new words beyond the learning sample, with precision as high as 92%.

Their work is based on the theory of *whole word morphology* [2], which seeks to account for morphological relations in a minimalist fashion. The central mechanism of the theory is the Word Formation Strategy (WFS), a sort of non-decomposable morphological transformation that relates full words with other full words and parses any complex word into a variable and a non-variable component. According to this, there is no distinction between inflectional and derivational morphology, and morphology is relational and not compositional. So, instead of looking for similarities between words (common stems, suffixes etc, as in all other approaches), they argue that words are defined by the differences amongst them, and it's some of these that constitute the domain of morphology. Hence, two words of a lexicon are meant to be morphologically related if and only if all the differences between them are found in at least one other pair of words of the same lexicon.

Their algorithm starts by comparing every word of a small lexicon and determines the segmental differences found between them. Merging the comparison structures that exceed some threshold in occurrence, they create a list of morphological strategies, which are then used to unify the words of the lexicon with some part of them. Preliminary results are encouraging, and the authors are looking forward to develop a more sophisticated sequence alignment routine, to

allow the handling of infixing, circumfixing or templatic morphologies (e.g. those of the Semitic type).

2.3.2 Obtaining affix inventories

A second, knowledge free category of research has focused on obtaining affix inventories.

Discover morphemes by using MDL

Brent, et al. [27] exploit the minimum description length (MDL) principle to discover the most powerful morphemic suffixes. They present two models for generating the words in a corpus, based on two linguistic hypotheses: (1) that morphemic suffixes and stems recombine with one another to form multiple words, and (2) that morphemic suffixes provide information about the syntactic categories of words formed with them. In such a fashion, words are generated by selecting a stem and a suffix from respective lists and then selecting a syntactic category from a list of such categories available to words with the selected suffix. Words are encoded into bit strings and form tables by the splitting of each word in a stem and suffix and then heuristics are applied to reduce the number of accountings in the search space. Their experiments output *-age*, *-al*, *-ed*, *-ing*, *-ion*, *-ity*, *-nce* and *-s* as the most common suffixes.

Creutz and Lagus also presented a more recent work [7] containing two methods for the unspervised segmentation of words into morphemes, one model using MDL and another one using Maximum Likelihood (ML) optimization. The experiments appeared successful, with the first model being slightly more efficient than the second.

Use of genetic algorithms

Something similar has been proposed by Kazakov [20], a combination of unsupervised and supervised learning techniques for the generation of word segmentation rules from a raw list of words, using genetic algorithms and inductive logic programming. The former provides a first approximation of the concept learnt and reduces the search space, while the latter learns rules that can be employed to segment unseen words. Results of their tests show that a set of rules for word segmentation can be learnt from a limited number of unannotated words (in the order of 10^3 words). The genetic algorithm in this work may be viewed as a search technique in the MDL framework for word segmentation.

Morphemes as concept for structures

DeJean [10] is inspired by the works of Zellig Harris [14], 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.

2.3.3 Performing a complete morphological analysis

Although the work presented in the previous categories does induce some information about a language's morphology, finding just the affixes of a language is not a complete morphological analysis of the specific language. Another knowledge-free category of research attempts is to induce a complete analysis of the morphology for each word of a corpus.

Guessing morphology

One approach that falls into this category is that of Jacquemin [17]. His algorithm is aimed not only at the derivation of affixes, but the automatic acquisition of morphological links between words in a corpus. These morphological relations actually rely on a measure of distance between two strings which is similar to the distances used for approximate string matching, following the notion that words which share a common stem are semantically related. Besides the text corpus, his algorithm also requires a list of multi-word terms.

The aforementioned measure of similarity is a comparison of truncations whose lengths are related to the lengths of the initial string (k-similarity). After words are related using this procedure, the algorithm looks for approximate equalities between multi-word terms and corpus utterances, using the idea of k-n-conflation. Next, conflations are grouped into classes and the incorrect ones are filtered out. The classification relies on a characterization of the morphological operation transforming the original term into its corpus occurrence. Finally, classes are clustered together if they are associated with similar morphological processes. Again, this step is based on some linguistic principles that form a measure of distance between the classes.

Linguistica

Goldsmith [13] proposes another method of minimum description length (MDL) analysis to model unsupervised learning of the morphology of European languages. Inspired by De Marcken's thesis on MDL [8], he attempts to provide both a list of morphemes and an analysis of each word in a corpus. His algorithm is implemented and named *Linguistica*, and is freely available on the Internet. He presents a set of heuristics that develop a morphological grammar and then uses MDL to determine whether the modifications proposed by the heuristics will be adopted or not, by eliminating inappropriate parses for every word in the corpus. Further information about Linguistica is presented in Section 2.4.

Knowledge-free morphology induction

An interesting and still quite straightforward approach is that of Schone and Jurafsky [35], who also propose an unsupervised morphology acquisition method, based on untagged corpus, which will be described in more detail. In general, their approach consists first of looking for potential suffixes by searching for frequent affixes, then looking for potential morphologically related pairs. A series of enhancements provide improved results, such as incorporation of semantic, orthographic, local syntactic information and transitive closure.

In more detail, their approach consists of the following states:

1. Identification of pairs of potential morphological variants
2. Determination of semantic vectors for each word
3. Correlation of the semantic vectors and creating of conflation sets
4. Augmentation with frequency information

5. Consideration of local context for part of speech info

6. Addition of words using transitive closure

In the **first phase**, beggining with an untagged corpus as input, they first generate a list of word pairs that might be morphological variants. One strong point in their algorithm is that it does not only seek to identify prefixes and suffixes, but circumfixes as well. From this lexicon that consists of all the words in the corpus, they identify and strip *pesudo-prefixes*, that is word begginings in excess of some threshold (T_1). The word residuals are inserted back into the lexicon as if they are valid words. Using this final lexicon, they now seek for suffixes, by inserting the lexicon into a trie and assembling a list of all trie branches that occur some minimum number of times (T_2). This list contains all the *potential suffixes*. In a similar manner, by reversing the ordering of the words, a list of *potential prefixes* is derived.

Additionally, a *potential circumfix* is a pair B/E where B and E occur respectively in potential prefix and suffix lists. The residual of a word after a potential circumfix is removed, is called a *pseudo-stem*. A potential circumfix that appears more than T_3 times is called a *candidate circumfix*. Following this, comes the derivation of a *rule*, which is a pair of candidate circumfixes sharing at least T_4 pseudo-stems. Two words sharing the same rule but distinct candidate circumfixes constitute a *pair of potential morphological variants* (PPMV). Finally, the set of all PPMVs for a common rule is a *ruleset*.

The final goal of the first phase is to find all the possible rules and corresponding rulesets. Several of these rules are quite valid, still there are some that are almost never valid, but the incorporation of semantics can help in determining the validity of each rule.

The **second phase** of the algorithm is where semantics are incorporated, by using the algorithm of Latent Semantic Analysis [9], which shows that valid semantic relationships between words and documents in a corpus can be induced with virtually no human intervention. This is done for the N most frequent words (because computations are expensive) and the remaining terms are "fold in" to a glob position.

Correlation of these semantic vectors takes place in **phase three**, by computing a score for a pair of words, namely the normalized cosine score NCS. If this score exceeds some threshold (T_5) for two words of a PPMV, then a valid relationship between the words is accepted.

In the **fourth phase**, affix and rule frequencies take part in the computation of the *orthographic* probability of validity that two words are morphological variants, motivated by the use of the minimum edit distance (MED), which is the minimum-weighted set of insertions, substitutions and deletions required to transform one word into another.

The authors also argue that there is no guarantee that two words that are morphological variants need to share similar semantic properties, so in **phase five** they incorporate the use of local syntactic contexts around words, in addition to the large-window contexts that were used in semantic processing. By use of *signatures* (collections of words that occur too many times around the word), they compute a syntax-based probability for each word. The result is that the probabilities of some low-score, but yet valid PPMVs increases because of their local context.

In the **sixth phase**, the algorithm tries not to deem PPMVs that still may seem unrelated, if they appear as members of other *valid* PPMVs. By combining

the probabilities of all independent (intermediate) paths from X to Z, they compute a branching probability, and when it exceeds T_5 the two are declared to be morphological variants of each other.

Evaluation of the algorithm on English, German and Dutch yield quite good results, resulting in almost a 20% relative reduction in overall induction error. A combination of such an approach with that of Yarowsky and Wicentowski [40] (which does very good work with irregular forms) could potentially lead to a quite successful derivation of a language's morphology.

Using orthographic and semantic similarity in another fashion

Another knowledge-free approach that exploits orthographic and semantic similarity is that of Baroni, et al. [26]. With an unannotated corpus as input, it returns a list of probable morphologically related word pairs. Orthographic similarity is measured using minimum edit distance (MED). Their approach however differs from that of Schone and Jurafsky [35] in that regarding orthography, they rely on the comparison between individual word pairs without requiring that the two pairs share a frequent affix. Also, from the point of view of semantics, they compute scores based on Mutual Information [6] instead of latent semantic analysis, looking at the co-occurrence patterns of the target words, rather than the similarity of their contexts.

Mutual Information between two words A and B is given by:

$$I(A,B) = log\frac{P_r(A,B)}{P_r(A)P_r(B)} \tag{2.3.1}$$

which intuitively means that the larger the deviation between the empirical frequency of co-occurrence of two words and the expected frequency of co-occurrence if they were independent, the more likely it is that the occurrence of one of the

two words is not independent from the occurrence of the other. The measurement of semantic similarity is thus based on the notion that related words will tend to often occur in the nears of each other.

Experiments with German and English inputs gave encouraging results, both in terms of precision, and in terms of the nature of the morphological patterns found within the output set.

2.4 The Linguistica Morphological Analyzer

In our work, we used the Linguistica system to perform morphological analysis for both the source and target languages. As mentioned in the previous section, Linguistica uses a set of heuristics to provide an initial morphological analysis. The first one (called take-all-splits), considers for each word of length l all the possible cuts into $w_{1,i}$, $w_{i+1,l}$, $1 \leq i < l$. For each cut, the metric H is computed (as seen in Eq. (2.4.1)) and the corresponding probability of the cut is given by Eq. (2.4.2), i.e.,

$$H(w_{1,i}, w_{i+1,l}) =$$

$$= -(i \log \text{freq}(\text{stem} = w_{1,i}) +$$

$$(l - i) \log \text{freq}(\text{suffix} = w_{i+1,l})) \tag{2.4.1}$$

where freq represents the number of times a stem or suffix appears in the corpus and

$$prob(w = w_{1,i}\ w_{i+1,l}) = \frac{1}{Z} e^{-H(w_{1,i}, w_{i+1,l})} \tag{2.4.2}$$

where the normalization factor Z equals

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

For each word, the best parse in the maximum likelihood sense is selected to bootstrap the heuristic and then the metric is optimized globally over all words, stems and suffixes in the corpus (usually the process converges after five iterations).

The second heuristic computes the counts of all sequences of characters with length n between two and six letters. Then for each n-gram $n_1, n_2 \dots n_k$ we compute the weighted mutual information metric:

$$\frac{[n_1, n_2, \dots, n_k]}{\text{Total count of } n\text{-grams}} log \frac{[n_1, n_2, \dots, n_k]}{[n_1][n_2] \dots [n_k]}$$

The top 100 scoring n-grams are kept and used to parse each word (if possible) into stem plus suffix. For those words that more than one splits exist, the previous heuristic is used to choose the best one.

Finally, for each stem the list of all corresponding suffixes is created which is referred to as a *signature*. Stems with the same suffix signatures are merged. Signatures that contain more than one stems and affixes are referred to as *regular signatures* and are of the form

$$\left\{ \begin{array}{l} \text{stem}_1 \\ \text{stem}_2 \\ \text{stem}_3 \end{array} \right\} \left\{ \begin{array}{l} \text{suffix}_1 \\ \text{suffix}_2 \end{array} \right\}$$

Heuristic rules are used to add stems or suffixes to regular signatures (based on similarities with other regular signatures) thus improving on the generalization power of the morphological rules. Note that the morphological signatures are derived in a fully unsupervised fashion.

2.4.1 Improving Linguistica

Linguistica offers the chance of adjusting some parameters in order to influence the resulting morphology, but in order to do this one must take into consideration

the way the morphology is built. A simple and cheap (in terms of time and computational power needed) way of increasing the precision of the resulting morphological analysis, on the expense of recall has been proposed by Giannis Klasinas in [21] and is presented in this section.

Examining the result of the morphological analysis that Linguistica provides, it can be observed that in most words that are mistakenly analyzed, the error is assigning stem characters to the suffix. The problem of false identification becomes more important when dealing with short words, where removal of the suffix usually leaves a very short stem which is possibly useless for training an SMT system. In order to overcome these problems, a heuristic rule is employed, which uses two parameters:

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

Every word that is analyzed by Linguistic is examined, and the analysis is kept only for words that have $l_{word} > l_0$ and $r_{word} < r_0$. In order to choose the appropriate values for these two heuristics, an experiment has been carried out. Linguistica was used to provide a morphological analysis based on a 1M token greek corpus, then 1k words were picked randomly (2k tokens). A human judge was then used to decide for each word if the analysis provided by Linguistica was correct or mistaken. The results for different values of l_0 and r_0 are shown in Table 2.1

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 2.1: Precision results for the additional heuristics in Linguistica

Chapter 3

The Baseline SMT System

3.1 Overview

In this Chapter the baseline SMT system that has been used in our experiments is presented. This system plays two different roles in our work. First, it is used as the lexical system with which the morphological SMT system is combined, resulting into the combined lexical-morphological SMT system that will be presented in the next Chapter. Second, it is used as the baseline to which we compare the new system, regarding their translation quality. This SMT system has been developed by Fanouris Moraitis [28]. Modification of the bilingual phrase extraction by Giannis Klasinas [21] has resulted into further improved translation quality results.

The rest of this chapter is organized as follows: Section 3.2 discusses the methodology used to address the problem of sentence boundary detection, while the original sentence alignment algorithm is presented in Section 3.3. Language modeling and translation modeling techniques that are used in the baseline system are discussed in Sections 3.4 and 3.5 respectively. Finally, the part of decoding is discussed in Section 3.6.

3.2 Sentence Boundary Detection

As discussed earlier in this document, Statistical Machine Translation systems use large bilingual (or multilingual) text corpora to train statistical models. These corpora are often not annotated or tagged in any way. The first step in pre-processing the texts in order for them to be used in the training process, is to pair sentences between the different languages, but in order for this to happen, there must be a way to detect where a sentence's boundaries are.

The process of sentence boundary detection may sound as a trivial issue, however this is not the case. The first thing that comes to mind is that a sentence ends with a period (.), but this is not always the case since the period is a character that is also used in acronyms or words like *e.t.c.* Other characters that often delimit the end of a sentence are *?*, *!* and ∴. In these cases it's more probable to have the end of a sentence when one of these characters are used since they don't tend to be used in other ways.

In order to address this problem, first the text is split into tokens. A token is considered to be a sequence of characters between two spaces. Then, every token that contains one of the characters of the set (. *!* : *?*) is further divided into three parts: a prefix, which consists of the characters before the special character, a candidate which is the special character itself and a suffix which is made of the rest of the characters that appear after the candidate. Additional information is also stored, as whether the token is some known abbreviation or honorific as well as whether the previous or next token begins with a capital letter. Each potential token that may be a sentence boundary is then examined by a function that applies a set of rules, using all the information gathered about the token, to conclude whether there should be a valid sentence boundary.

3.3 Sentence alignment

After the sentence boundaries in the texts have been detected, the sentences that correspond to one another in the parallel texts must be found, hence the sentences must be aligned. The method that has been used for the sentence alignment is based on the assumption that the length of a sentence in one language is often proportional to the length of the sentence in the other language too, i.e. long sentences in one language usually correspond to long sentences in other languages.

The distance between two sentences is defined as $-\log P(\text{match}|\delta)$ where δ is a random variable that depends on the lengths of the two sentences, l_1 and l_2. Specifically, $\delta = \frac{l_1 - l_2 c}{\sqrt{l_1 s^2}}$, where c represents the mean of the model and s^2 it's standard deviation. These values are computed empirically.

Using Bayes rule then, $P(\text{match}|\delta)$ can be written as:

$$P(\text{match}|\delta) = \frac{P(\delta|\text{match})P(\text{match})}{P(\delta)}$$

where $P(\delta)$ can be ignored since it is the same for all possible matching pairs. The value of $P(\text{match})$, which is the prior probability is given from Table 3.1. The type of a match means whether there is a 1:1 sentence match, a 1:0 or 0:1 where a sentence in one language does not appear in the other language, and 1:2 or 2:1 where one sentence in the first language occurs as two sentences in the second language and vice versa.

Match type	$P(\text{match})$
$1-1$	0.89
$1-0$ or $0-1$	0.005
$1-2$ or $2-1$	0.0445

Table 3.1: Match prior probabilities

Since δ follows the normal distribution with mean $c = 0$ and standard deviation $s^2 = 1$ we can assume that $P(\delta|\text{match}) \approx P(\delta)$. So, suppose there is a match in the range $[-\delta_0, \delta_0]$ we have:

$$P(\delta|\text{match}) = P(-\delta_0 < \delta < \delta_0)$$

and using the Cumulative Distribution Function we come up with

$$P(-\delta_0 < \delta < \delta_0) = 2[1 - P(\delta < \delta_0)]$$

where

$$P(\delta < \delta_0) = \frac{1}{\sqrt{2\pi}} \int_{-\inf}^{\delta_0} e^{\frac{-\delta^2}{2}}$$

so finally we come up with

$$P(\text{match}|\delta) = P(\text{match}) \left[2 \left(1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\delta_0} e^{\frac{-\delta^2}{2}} \right) \right]$$

Every possible pair of sentences is assigned a cost that is computed in the way described above, then dynamic programming is used which results into the best matches between the sentences. This method of sentence alignment is an implementation of the algorithm presented in [36].

3.4 Language modeling

As we have seen earlier, SMT is based on the maximization of the product of two probabilities which model fluency and faithfulness. Fluency is modeled using the language model, which computes the prior probabilities of each phrase, based on occurrences in the training corpus. This corresponds to the prior probability of a word of phrase to occur in a language.

For the calculation of these probabilities n-grams are used, which are phrases of n words and are called unigrams, bigrams, trigrams etc when they use one,

two or three words and so on respectively. In order to model the sequence of two words, hence a bigram, the number of occurrences of the phrase is divided by the number of occurrences of the first word. For example, for the phrase *he speaks* the probability $P(\text{speaks}|\text{he})$ is calculated as:

$$P(\text{speaks}|\text{he}) = \frac{\#(\text{he speaks})}{\#(\text{he})}$$

For the computation of these models, we used the SRILM toolkit [1] which is available over the Internet for non-commercial applications. Our implementation used fourgrams.

3.5 Translation modeling

Quantification of the faithfulness of the translation is achieved by using a translation model, as we have seen before. The translation that can be modeled may be word-to-word or phrase-to-phrase. Our baseline system used phrase-based translation which uses the results of a word-based translation model.

3.5.1 Word-based translation modeling

The technique used to achieve word-based translation, first translates the words of a sentence to the target language and then re-orders the words. In some cases, a word may be translated into two or more other words, that may exist in sequentially or scattered in the sentence. To model these variations, the notions of fertility and distortion are employed. *Fertility* is the probability that a word in the source language will be translated into k words in the target language, while *distortion* is the probability that a word in position p_n in the sentence n of the source language will produce a word in position p_f in sentence f in the

target language. Also, the notion of spurious words is used: it is assumed that in position zero of every sentence in the source language the word *NULL* exists, which can give spurious words in sentence f with probability p_1.

For the translation modeling the IBM Model 3 has been chosen [5]. The process begins with the input sentence, written in the source language being transformed first by examining the fertility of each word and creating or deleting words according to them. Spurious words are then inserted and then the words are translated to the target language. Finally, the words are re-ordered using the distortion probabilities.

In order to produce the word alignment, we used the GIZA++ toolkit [31], an extension of the program GIZA (part of the SMT toolkit EGYPT). GIZA++ takes as input the sentence-aligned bilingual texts and computes the most probable word alignment for every sentence. This alignment is then used to calculate phrase-level probabilities between the source and target languages.

3.5.2 Phrase-based translation modeling

Word-based SMT systems are simple systems that do not take advantage of the contextual information of the texts, since they translate one word at a time. It is clear that if we could model the translation between phrases, instead of words, the translation quality would improve. In order to build such phrase-based translation models, we need to extract the bilingual phrases from the alignments provided by GIZA++.

A bilingual phrase may be defined as a pair of m source words and n target words. In order to extract these phrases from a word aligned training corpus, two additional constraints are posed: the words should be consecutive and consistent with the word alignment matrix, which means that the source words are aligned

only to the target words and vice versa. Phrases or words that are rejected are not ignored but regarded as being created from the word *NULL*.

The resulting bilingual phrases result in very large files, but since they don't have to be long, only phrases of up to three words length are kept. For the phrases that are kept, the translation probability is calculated as:

$$P(\bar{f}|\bar{n}) = \frac{\text{count}(\bar{f}, \bar{n})}{\sum_{\bar{f}} \text{count}(\bar{f}, \bar{n})} \tag{3.5.1}$$

3.6 Decoding

As it has been discussed in Section 1.2.5, after the translation model has been built, the best translation needs to be computed. For this task, we used the Pharaoh decoder [23], a beam search decoder for phrase-based statistical machine translation models available freely for non-commercial purposes. When supplied with the translation and language models that have been computed with the tools presented above, the decoder computes the best (most probable) translation from the source to the target language.

Pharaoh begins the search in an initial state where no source language input words are translated and no target language words are generated. New states are created by extending the target language output with a phrasal translation that covers some of the input words not yet translated. The current cost of each new state is that of the original state multiplied by the translation, distortion and language model costs of the added translation. Each hypothesis is represented by a back link to the previous best state (for back-tracking), a coverage vector, the last two words generated (in order to compute future language model costs), the end of the last phrase covered, the last added phrase, the cost so far and an

estimate of the future cost. Final states are hypotheses that cover all words in the input sentence, and the one with the lower cost (higher probability) is selected as the best translation.

Except for computing the best translation, Pharaoh can also be instructed to output the lattice of the probable paths that have been used. Running the decoder with the appropriate options allow for the creation of bigger lattices (in the expense of speed). We used these lattices to implement our lexical-morphological SMT system, as will be described in the next sections.

Chapter 4

Morphology Incorporation into SMT

4.1 Overview

SMT systems have proven to be a valuable tool for the unsupervised, automatic translation of texts between two languages. As we have seen in Chapter 1, these systems can be trained from untagged, parallel bilingual (or multinlingual) corpora of text documents translated in two (or more) languages. The quality of the produced translations of such systems may be evaluated either by human experts, or automatically, by computing metrics that quantify how fluent and faithful a given translation may be. These metrics are presented in Chapter 5.

All of the MT approaches that have been presented in Section 1.1.3 employ little or no linguistic knowledge at all. They operate on the lexical level without taking advantage of the linguistic information underlying in the texts. This linguistic information has the potential of improving the performance of SMT systems, especially when limited amounts of parallel training data sets are available.

Our work proposes a method for the improvement of existing, lexical SMT systems by the incorporation of morphological knowledge into such systems, and

specifically information about word-stems resulting in what we call morphological SMT systems. Such a system is then combined with the traditional lexical SMT system. This combination has proven to improve translation quality significantly, as it will be demonstrated in Chapter 5. Moreover, the process is fully automated and unsupervised; first, the morphological information is extracted automatically from the corpora, using a robust version of the unsupervised morphology acquisition algorithm that has been presented in Section 2.4. Second, the stems are incorporated into the SMT system using a generic statistical framework which combines a word-based and a stem-based SMT system [19].

The proposed SMT system implementation uses information about stems of words, discarding the information that can be obtained by the affixes. The morphological SMT system then operates at the stem level, performing translation between stems of one language to another. However, affix information could enhance the produced translation, since discarding it inevitably results into some loss of information. In order to exploit this information, we propose an extension to the statistical framework of the combined lexical-morphological SMT system we have implemented. Since this part of the incorporation has not yet been implemented, more information about it is displayed in Section 6.1.

The rest of this Chapter is organized as follows: Section 4.2 describes the motivation that drove to the idea of the morphological knowledge incorporation, while the way such an incorporation addresses the problem of sparse training data is discussed in Section 4.3. The proposed system architecture is illustrated in Section 4.4 while the mathematics that form the statistical framework on which the incorporation is based are presented in Section 4.5. An implementation of the combined lexical-morphological system is described in Section 4.6. Finally, a pointer to the extension to the statistical framework for the incorporation of

affix information is given in Section 4.7.

4.2 Motivation

The basic idea behind Statistical Machine Translation is briefly the computation of probabilistic models about how a word in one language is translated into some word in another language. Phrase-based SMT has provided further improvement in the quality of the translation, by modeling relationships between phrases, instead of plain words. The translation model of a phrase-based SMT system computes the probability of a given phrase, in a source language, to be translated in another phrase in the target language.

Both word-based and phrase-based SMT methodologies operate in the lexical level. However, as we have seen in Chapter 2, words are not monolithic constructs of symbols, but are usually built by the combination of smaller units, morphemes. Studying the morphology of a language reveals a vast amount of information about how words are created from simple morphemes, according to some inherent rules, and how each word that is generated by the same stem is used in different occasions. All this information is lost in today's SMT systems that operate on the lexical level.

Although morphological knowledge contains a lot of useful information about the formation of words, it is not easy for it to be incorporated into SMT systems. A morphological analysis that results into rules about word formation could be incorporated in a rule-based MT system, but is not very useful in SMT since the latter uses statistical means to model the translation between two different languages. In order for it to be exploited by SMT systems, there has to be a way of incorporating such knowledge into the SMT framework.

4.3 Addressing data sparseness problems

An application where morphological knowledge may prove to be quite useful is in the case of sparse training data. The heart of the SMT methodology is the computation of the translation model, which is based solely on the training data that are available. The size of the training data is important in building better translation models; the bigger the data that has been used to train the model, the better the probabilities computed. However, probabilities are computed for every word, or phrase, hence they depend on how often the words appear in the corpus. Even a large training corpus may have quite sparse data, i.e. distinct words may not appear very often in it.

Using morphological knowledge that has been obtained, the data can become more "dense". If the training corpora were stemmed, then every distinct stem would appear many more times than a single word that is derived from it does, so a translation model trained on such corpora would be better trained. This gives a definite advantage over the translation of words that are rare derivations of a common stem, since they would appear very little (or not at all) in the training data. However, since the stem of the word would probably appear many times in the stemmed training corpus, it's translation to the corresponding stem of the target language could be done quite successfully. The resulting stem would then have to be converted to the proper word in the target language, possibly by using the information of the affixes of the original word.

4.4 System Architecture

The procedure that has been briefly described is equivalent to a word-to-word translation system that performs the translation through a stem-to-stem SMT

system. We call such a system a morphological SMT system, since it performs translation exploiting the inherent morphological knowledge in the texts. This system could then be combined with the simple lexical SMT system, resulting in what is called a combined lexical-morphological SMT system.

The morphological SMT system that we propose consists of three different modules:

- the morphological analyzer
- the stem-to-stem SMT system
- the morphological generator

The same input is provided to both the lexical and the morphological system. In the case of the lexical system, translation takes place and the output is produced. For the morphological system, the input text is first analyzed by the morphological analyzer, resulting in a set of a stem and affixes for every word. The stems are then translated using the stem-to-stem SMT system and the morphological generator converts the produced stems into proper words. The output of this system is then combined with the output of the lexical system. More about how this combination takes place is discussed in Section 4.5.4.

The architecture of the combined lexical-morphological SMT system is illustrated in Figure 4.1. In this figure the symbols W_s and S_s denote words and stems in the source language, while W_t and S_t refer to the words and stems of the target language respectively.

4.4.1 Morphological Analyzer

As stated already, a morphological analyzer is responsible for splitting a word into it's stem and affixes, after performing an analysis to it. Such an analyzer may be

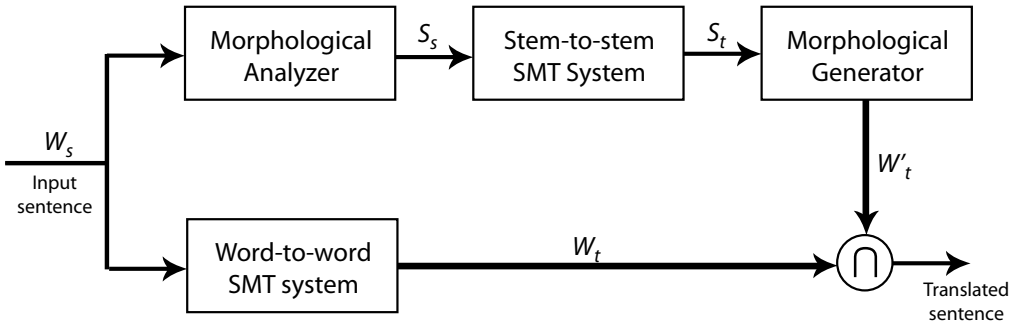

Figure 4.1: System Architecture

implemented in different ways, depending on the level of analysis it performs, i.e. it may perform a complete morphological analysis or simply stem the words. In essence, the morphological analyzer computes the probabilities $P(S|W)$ where S denotes the stem of the word W.

In our work we used the Linguistica morphological analyzer that has been presented in Section 2.4. Running Linguistica over an original training corpus, we obtain a list of stems and the affixes that they may be combined with, resulting into different words of the same stem. This information is then consulted, in order to split a word into it's stem and suffix. We consider this procedure to be deterministic, i.e. there is only one way to stem a word. This simplifies the model since for every word we will have $P(S|W) = 1$.

4.4.2 Stem-level SMT system

The stem-to-stem SMT system can be implemented like any traditional SMT system, with the difference that the training of the language and translation

models has been performed on stemmed corpora. In the training phase, the texts of the training corpora are processed by the morphological analyzer which produces their stemmed versions, which are then used to train the language and translation models of the stem-to-stem SMT system.

4.4.3 Morphological generator

A morphological generator is capable of performing the reverse process than that of an analyzer; given a stem, it has the duty of finding the words that are derived from it. Obviously, there are many words that can be derived from a single root, so choosing the appropriate word is hard. The morphological generator thus computes the probabilities $P(W|S)$.

A first approach is that the generator produces every possible word that may be derived from a specific stem. This approach may sound naive, since it is almost definite that the wrong words will be produced in this step. However, since the output of this phase will be combined with that of the lexical SMT system, all invalid words that have been produced will be "discarded" by the lexical system, because of the language model it uses.

The second approach is that of a probabilistic model, where every word that can be derived from the specific stem is assigned a probability. These probabilities may, in the simplest case be computed in a maximum likelihood fashion, i.e. depending on how often every word occurs in the training corpus. A more sophisticated word production algorithm though, would take into consideration the original affixes that have been discarded in the process of stemming, so to apply the proper affixes to the stem and thus turn it into a word. More about this approach is discussed in Section 6.1.1.

4.5 Statistical framework

This section describes the underlying mathematics that form the statistical framework on which morphology knowledge incorporation may take place.

4.5.1 Lexical SMT system

The traditional lexical SMT system that is used for the combined SMT system, is based on the original theory of SMT. As we have seen in Section 1.2, the goal of translation in traditional SMT systems is a posterior probability maximization problem. Slightly changing the notation now, let us consider W_s and W_t to be word sequences for the source and target languages respectively. Then, the problem of translation can be formulated as:

$$\hat{W}_t = \arg\max_{W_t} P(W_t|W_s) \tag{4.5.1}$$

where $\hat{W}_t$ is the translated sequence of words in the target language. We are no longer using the terms native and foreign language since translation can occur in either ways; instead, we consider the source language to be the one that the input sentence in written in, and the target language the one that we wish to translate the input sentence into.

4.5.2 Stem to stem translation

Let us now consider S_s and S_t to be sequences of stems for the source and target languages respectively. These sequences of stems have come up from the morphological analysis that has been described in the previous section. A stem-to-stem machine translation can then be formulated as:

$$\hat{S}_t = \arg\max_{S_t} P(S_t|S_s) \tag{4.5.2}$$

This results into a system that operates at the stem level, which can be trained from the stemmed training corpora, as described in Section 4.4.2.

4.5.3 Word-to-word translation via the stem-to-stem system

Having the statistical models for the morphological analyzer and generator, as well as the translation model for the stem-to-stem system, we can achieve a word-to-word translation based on the stem-to-stem system. Rewriting Eq. (4.5.1) we have:

$$\begin{aligned}
\hat{W}_t &= \arg\max_{W_t} P(W_t|W_s) \\
&= \arg\max_{W_t} \sum_{S_t,S_s} P(W_t, S_t, S_s|W_s) \\
&= \arg\max_{W_t} \sum_{S_t,S_s} P(W_t|S_t, S_s, W_s)P(S_t|S_s, W_s)P(S_s|W_s) \\
&= \arg\max_{W_t} \sum_{S_t,S_s} P(W_t|S_t)P(S_t|S_s)P(S_s|W_s) \qquad (4.5.3)
\end{aligned}$$

provided that W_t, S_s are conditionally independent given S_t; W_t, W_s are conditionally independent given S_t, S_s; and W_s, S_t are conditionally independent given S_s. This equation corresponds to a word-to-word translation model which is performed via the stem-to-stem system, i.e. $W_s \rightarrow S_s \rightarrow S_t \rightarrow W_t$.

Eq. (4.5.3) can be further simplified as follows: the mapping $S \rightarrow W$ is a many to one mapping and $P(S_s|W_s) = 1$, because the mapping $W_s \rightarrow S_s$ is deterministic, so the double summation at Eq. (4.5.3) becomes a single summation over S_t only, as follows:

$$\hat{W}_t = \arg\max_{W_t} \sum_{S_t} P(W_t|S_t)P(S_t|S_s) \qquad (4.5.4)$$

We refer to this SMT system that translates from W_s to W_t via an intermediate morphological representation S, as a morphological SMT system. Implementation of such a system is now easy; as it can be seen from Eq. (4.5.4), one must compute the probability model $P(W_t|S_t)$, which is the morphological generator that has been discussed before, as well as the probability model $P(S_t|S_s)$ which is the stem-to-stem translation model.

4.5.4 SMT system combination

Once the morphological SMT system is built, it is combined with the traditional lexical SMT system. This combination may assume that each system computes probabilities independently of each other, i.e.,

$$\hat{W}_t = \arg\max_{W_t} [P(W_t|W_s)]^{w_0} \left[\sum_{S_t} P(W_t|S_t) P(S_t|S_s) \right]^{w_1} \tag{4.5.5}$$

where w_0 and w_1 are weights that model the "confidence" we have in each translation the lexical and the morphological SMT systems provide correspondingly. The combination of these two systems is performed in order to overcome the weakness of the conditional independence assumptions of Eq. (4.5.3). This combination may be performed at an early, or late stage.

Experimentation on the values of the weights w_0 and w_1 may boost the translation performance. The results of our experiments in regard of these weights are presented in the next Chapter.

4.6 Combined Lexical-Morphological system implementation

This section discusses our implementation of such a combined lexical-morphological SMT system. Our approach uses late integration and lattice re-scoring. The notion is to use the lattices that are produced by the Pharaoh decoder, in the stage of decoding, for both the lexical and the morphological SMT systems. These lattices start from an initial empty state and create nodes for every possible translation, as described in Sections 1.2.5 and 3.6.

Both lattices are read and then represented as weighted finite state machines (FSMs). In our work, we used the AT&T FSM Library [25] for the representation of finite state machines and the operations applied to them (closure, composition, intersection, best path decoding). The AT&T FSM Library is a set of general-purpose software tools for building, combining, optimizing, and searching weighted finite-state acceptors and transducers.

The lattice that has been produced by the SMT system is the output of the second phase of the stem level translation system. The next step is to pass it through the morphological generator. In our implementation, the generator simply produces all the words that can be derived from the stem, using the sets of stems and affixes that have been provided by Linguistica in the training process. This is implemented as the composition of the FSM that represents the lattice and a new FSM that is constructed, containing identity mappings from a stem to every possible word. All these identity mappings are assigned the same fixed cost, making them all equiprobable. The reason for making such a loose word production is that invalid words will be discarded in the next step of the two systems combination.

The next step in the procedure is the combination of the two systems. The two FSMs that have resulted from the procedure till now, are weighted and then intersected to obtain a final FSM, which implements the output of the combined lexical-morphological SMT system. This FSM is then decoded in order to find the best path, which is the best possible translation of the input sentence into the target language.

All the steps that have been mentioned in the procedure just described, may be outlined as follows:

1. The lexical SMT system that computes the probabilities $P(W_t|W_s)$ is built. This is the translation model for the pure lexical SMT system that is being constructed using the baseline system described before.

2. The Linguistica morphological analyzer is given a set of training data, in order to perform a morphological analysis and output the rules that govern how a word is separated into its stem and affixes.

3. The training corpus is stemmed using the unsupervised rules derived from the Linguistica morphological analyzer in the previous step. The new stemmed corpus is stored, in order to be processed for the construction of the stem-level SMT system.

4. The stemmed corpus is used to derive the morphological (stem) SMT system that computes the probabilities $P(S_t|S_s)$, as has been described in Section 4.5.2.

5. Every sentence in the evaluation corpus is decoded using the lexical SMT system producing a lattice of possible word-level translations. This lattice is then represented as a finite state acceptor F_W.

6. Every sentence in the evaluation corpus is stemmed (again using the unsupervised rules that have been derived in step two) and then decoded using the morphological SMT system. The resulting lattice contains all possible stem-level translations and is represented as a finite state acceptor F_S.

7. The stem to word model $P(W_t|S_t)$ in the target language is constructed by running the Linguistica system on the target language corpus and obtaining the morphological signatures. The stem to word model is represented as a unweighted (costless) finite state transducer T_{SW}, i.e., in our case, we assume that all possible words that can be generated from a stem are equiprobable [1].

8. The stem acceptor F_S and the stem to word transducer T_{SW} are composed to obtain a stem-to-word mapping; the resulting transducer is projected to its output symbols to obtain the finite state acceptor $F_{W'}$.

9. F_W and $F_{W'}$ acceptors are re-weighted[2] (weights multiplied) by the factors w_0 and w_1 as discussed in Section 4.5.4. (in practice, we don't weight F_W and w_0 is always 1.).

10. The weighted acceptors F_W and $F_{W'}$ are intersected and the best path of the intersection is found using Viterbi decoding. The best path T' represents the translated sentence of the combined lexical-morphological SMT system.

The aforementioned process can be formulated as follows:

$$T' = \text{bestpath}\{([F_S \circ T_{SW}]_2 * w_1) \cap F_W\}$$

[1] In order to guarantee non-empty composition in the next step, all words contained in F_W and F_S were added as identity mappings in T_{SW} and then Kleene closure was applied to T_{SW}.

[2] In our implementation the tropical semiring was used, so the information in the FMSs is cost instead of probability.

where $\circ$ represents composition, $\cap$ intersection, $*$ weighting and $_2$ projection to the output symbols; T', F_S, T_{SW}, F_W and w_1 are defined above.

The implementation that has just been described slightly differs from Eq. (4.5.5), in that the summation over S_t is substituted by maximization. Hence, our implementation approximates Eq. (4.5.5) as follows:

$$\hat{W}_t = \arg\max_{W_t} [P(W_t|W_s)]^{w_0} \left[\max \left[P(W_t|S_t)P(S_t|\arg\max_{S_s} P(S_s|W_s))\right]\right]^{w_1} \tag{4.6.1}$$

Such a simplification could create probability normalization problems, but is acceptable since the mapping from W_s to S_s is deterministic and the distribution $P(W_t|S_t)$ has a low entropy. Even if this simplification did not take place and the summation was implemented, there are no guarantees that all instances of S_t that can generate W_t are available in the lattices. The substitution of maximization for summation in Eq. (4.5.5) results in simplifying the model computationally.

The system was implemented in Perl 5 scripts, using the tools that have been described in the previous Chapter when necessary.

4.7 Incorporating affix information

Discarding affixes in the process of the morphological analysis results in information loss. In the system described so far, only the stems are kept, translated into the target language and then converted back into words. It is possible though to incorporate the information that affixes contain, to further enhance the translation quality of the system, by creating an affix-to-affix translation system and then incorporating it into the statistical framework presented so far. Such an incorporation of affix information has not been implemented in the SMT system we have built and is presented in Section 6.1.1.

Chapter 5

Experimental Results

5.1 Overview

This Chapter presents our experimental results in testing the combined lexical-morphological SMT systems that we have implemented. The comparison criterion is translation quality, a criterion which is rather subjective for human judges. Because of this, we have not evaluated our results with human judges but using two metrics that are used in machine translation evaluation and are presented in Section 5.2.

The comparison is made between two systems: the traditional, lexical SMT system which is the baseline; and the combined lexical-morphological SMT system that has been described in the previous Chapter. Several experiments have been carried out, varying in different training corpus sizes as well as combination weights. To further validate our results, we have performed significance tests to verify that the improvement that has been observed is indeed significant. For the purpose of statistical significance tests, we used the method of bootstrap resampling.

The rest of this Chapter is organized as follows: the automatic evaluation metrics that are used are introduced in Section 5.2 while information about training

the morphological analyzer is presented in Section 5.3. The corpora that has been used are discussed in Section 5.4 while Section 5.5 presents the parameters of our experiments. The results of the experiments are illustrated in Section 5.6 while the statistical significance background and test results are displayed in Section 5.7.

5.2 Evaluation Metrics

In Section 1.2.3, we discussed about two different criteria that define how good a translation is: faithfulness and fluency. The relationship between these criteria as concepts and the underlying mathematics of SMT is a direct one; faithfulness is achieved through the translation model, and fluency through the language model. Maximization of these two probabilities results in the best possible translation. These two criteria, faithfulness (or *adequacy*) and fluency are the main criteria in machine translation evaluation.

Assigning the task of translation quality evaluation to human judges, is a difficult task mainly for two reasons; first, it is expensive, since it is a laborious and time-consuming process; second, because it is a process rather subjective, so the outcome of the evaluation could be biased when using different judges.

In order to overcome these limitations, two automatic scoring metrics have been devised and are widely used in MT evaluation: *BLEU* [34] and *NIST* [18]. The system output is compared to a reference translation of the same source text, then a value for these two metrics can be computed deterministically.

5.2.1 The BLEU metric

BLEU was the first metric that has been devised to quantify the translation quality of MT systems. It is based on the modified n-gram precision, which counts how many n-grams of the candidate translation match with n-grams of the reference translation. Given the precision p_n of n-grams of size up to N (usually $N = 4$), the length of the test set in words (c) and the length of the reference translation words (r) then BLEU is defined as:

$$\text{BLEU} = \text{BP} \exp(\sum_{n=1}^{N} \log p_n)$$

where

$$\text{BP} = \min(1, e^{1-\frac{r}{c}})$$

Computation of p_n is done by first counting the maximum number of times an n-gram occurs in any single reference translation, clipping the total count of each candidate n-gram by its maximum reference count, adding these clipped counts up and dividing by the total number of candidate words. BP is termed the *Brevity Penalty* which accounts for the factor of recall and penalizes candidates shorter than their reference translations, so the system can not only translate fragments of the test set of which it is confident, resulting in high precision. The final BLEU score is the geometric average of the modified n-gram precision multiplied by the brevity penalty.

5.2.2 The NIST metric

The NIST scoring metric has been based on BLEU and has been proposed in order to address some limitations of it. First, because the BLEU metric uses a geometric mean over N the score is equally sensitive to proportional differences in

co-occurrence for all N. As a result, there exists the potential of counterproductive variance due to low co-occurrences for the larger values of N. An alternative would be to use an arithmetic average of N-gram counts rather than a geometric average.

Second, it proposes that it would be better to weight more heavily those n-grams that are most informative, i.e. the ones that occur less frequently. This would also help to combat possible gaming of the scoring algorithm, since the n-grams that are most likely to (co-)occur would add less to the score than less likely n-grams.

In order to capture the difference in the information that different n-grams carry, information weights are introduced as follows:

$$Info(w_1 \cdots w_n) = log_2 \left(\frac{\text{\# of occurrences } w_1 \cdots w_{n-1}}{\text{\# of occurrences } w_1 \cdots w_n} \right)$$

then the formula for the score calculation is:

$$Score = \sum_{m=1}^{N} \left\{ \frac{\sum_{W^1} Info(w_1 \cdots w_n)}{\sum_{W_2}(1)} \right\} \exp \left\{ \beta \log^2 \left[\min \left(\frac{L_{sys}}{\bar{L}_{ref}}, 1 \right) \right] \right\}$$

where W^1 denotes the set of all w weights that co-occur and W^2 the set of all w weights in the system output; β is chosen to make the brevity penalty factor 0.5 when the number of words in the system output is $2/3^{rds}$ of the average number of words in the reference translation; $\bar{L}_{ref}$ is the average number of words in a reference translation, averaged over all reference translations and L_{sys} is the number of words in the translation being scored.

5.3 Morphological analysis

As discussed in the previous chapter, we used the Linguistica morphological analyzer to automatically derive morphological rules. These rules have been derived

from a 5M word parallel translation corpus, in both the source and target languages, in an unsupervised way. As we have discussed in Section 2.4, Linguistica uses a set of heuristics to develop a probabilistic grammar and then depends on Minimum Description Length analysis to determine which of the rules proposed will be adopted. In the same section is has been also shown that precision increases by introducing two heuristics, length of word l and suffix ratio r. During our experiments the values that have been used are $l_0 = 6$ and $r_0 = 0.4$.

5.4 Corpora

The systems that have been implemented have been trained on parts of the Europarl corpus [33], a parallel corpus in 11 European languages which is extracted from the proceedings of the European Parliament. The training corpus sizes that have been used are presented in Section 5.5.1.

The test sets have also been drawn from the Europarl corpus. Care was taken when creating training and test sets: it is common for sequential segments of texts in the corpus to share the same vocabulary and style, so it is better to avoid creating models based on such data. In order to overcome this, we used broad sampling, so the chosen data is evenly distributed in the corpus. Both systems have been trained on the same training sets.

5.5 Parameters

Experimentation in the task of comparing the translation quality of the two systems has been done by altering two parameters:

- the training corpus size

- the weights used in the systems combination

5.5.1 Parameter 1: Training corpus size

The first parameter, training set size, directly affects the quality of the output of an SMT system. The larger the training data, the better the translation quality since the models are better trained. Also, as the training set becomes larger, words tend to appear more often, while in systems that have been trained with a small corpus some words may not appear at all.

We expect that the combined lexical-morphological SMT system will provide great improvement in systems that have been trained with small corpora and that this improvement will become more subtle as the systems get better trained. Experimenting on different training set sizes lets us evaluate in which cases the morphological knowledge incorporation truly boost the system's performance.

For our experiments we used three different training corpora sizes, of 800k, 2.5M and 4.5M words each. For every set, the models of both the lexical and the combined SMT system were trained and a series of experiments took place by changing the second parameter, the weight w_1. Tables 5.1 and 5.2 summarize some important information about the training sets used, for the english and greek language respectively[1].

words	distinct words	distinct stems	stem:word ratio
800k	21k	16k	76%
2.5M	35k	27k	77%
4.5M	45k	36k	80%

Table 5.1: English training corpus information

[1]The values presented in the table are approximate values

words	distinct words	distinct stems	stem:word ratio
800k	45k	33k	73%
2.5M	76k	55k	72%
4.5M	100k	75k	75%

Table 5.2: Greek training corpus information

Examining the values of these tables leads to some interesting realizations. First, it is evident that for the same corpus, the distinct english words are almost half of the greek ones, due to the fact that the greek language has a richer vocabulary than the english one. The same analogy applies to the distinct stems found for every language as well, which draws a second point, that the stem-to-word ratio is roughly the same for both english and greek with an average of 75%. This means that after stemming the corpus, the vocabulary drops to the 75% of the initial size.

One other interesting fact is that the increase of the distinct stems is not proportional to the increase of the corpus size. This is expected, since the number of stems that exist in some language is somewhat fixed, so increasing arbitrarily the training set does not necessarily mean that the vocabulary should increase as well.

5.5.2 Parameter 2: Combination weights

The second parameter in our experiments regards the combination weights that are used in the system combination. Specifically, since $w_0 = 1$ we conducted several experiments by altering the weight w_1. It should be noted that since in our implementation we are using costs instead of probabilities, a smaller value for w_1 gives more power to the combined SMT system in the combination.

5.6 Results

This section illustrates the results of our experiments. For the sake of illustration we have named the baseline SMT system as system A and the combined lexical-morphological SMT system as system B. Since we're interested in the improvement that the new SMT system has over the baseline, we have calculated the differences in the NIST and BLEU scores between the two systems, by subtracting the scores of the baseline from the scores of the new system, as follows:

$$NIST_d = NIST_B - NIST_A$$

$$BLEU_d = BLEU_B - BLEU_A$$

where $NIST_A$ and $NIST_B$ are the NIST scores for systems A and B respectively and $NIST_d$ is the difference between the two scores. A positive value for $NIST_d$ thus means the new systems has improved the translation quality. The same applies for the BLEU scores.

As the same input is given to both systems, some sentences produce different translations and some not. In order to evaluate the actual improvement in translation quality, we focus on the sentences that did result in different translations. This allows for an evaluation of the improvement that the morphological incorporation provides over the baseline system. Both systems have been provided with a test set of approximately 40k sentences. The sentences that resulted in different translations were found to be approximately 8k, which stands for roughly 20% of the original test data. These sentences have formed the true test set on which the evaluation metrics have been calculated.

Figures 5.1 and 5.2 illustrate how different values of the w_1 parameter (x-axis) affect the scores of both metrics for the three different training sets. Also,

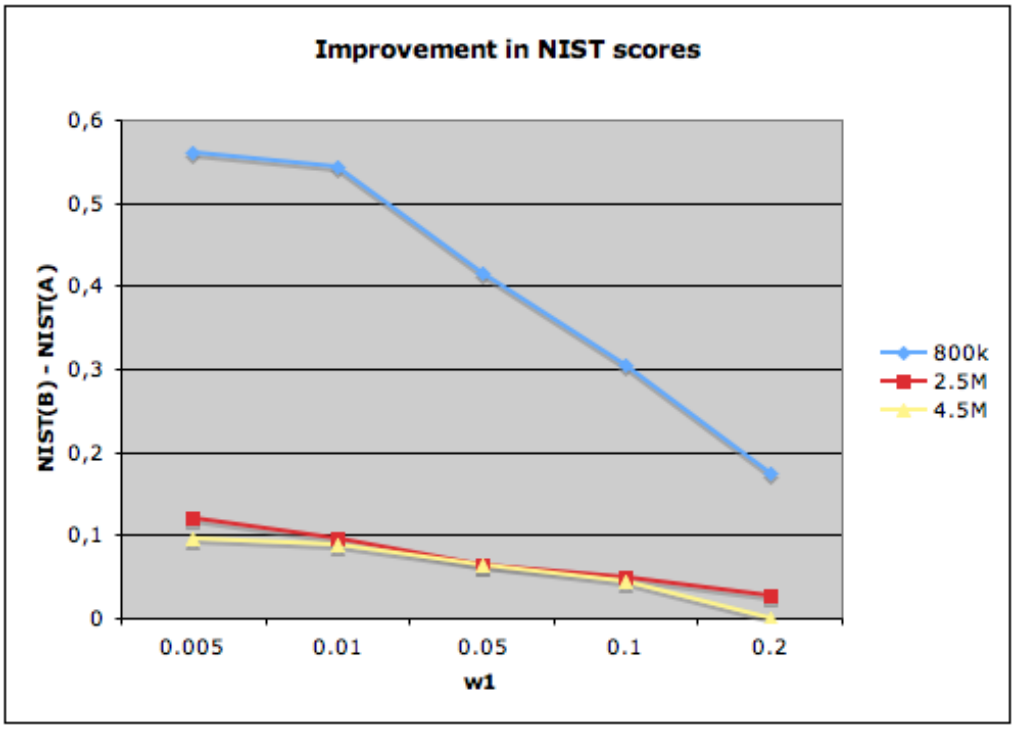

Figure 5.1: Improvement in NIST scores (over w_1)

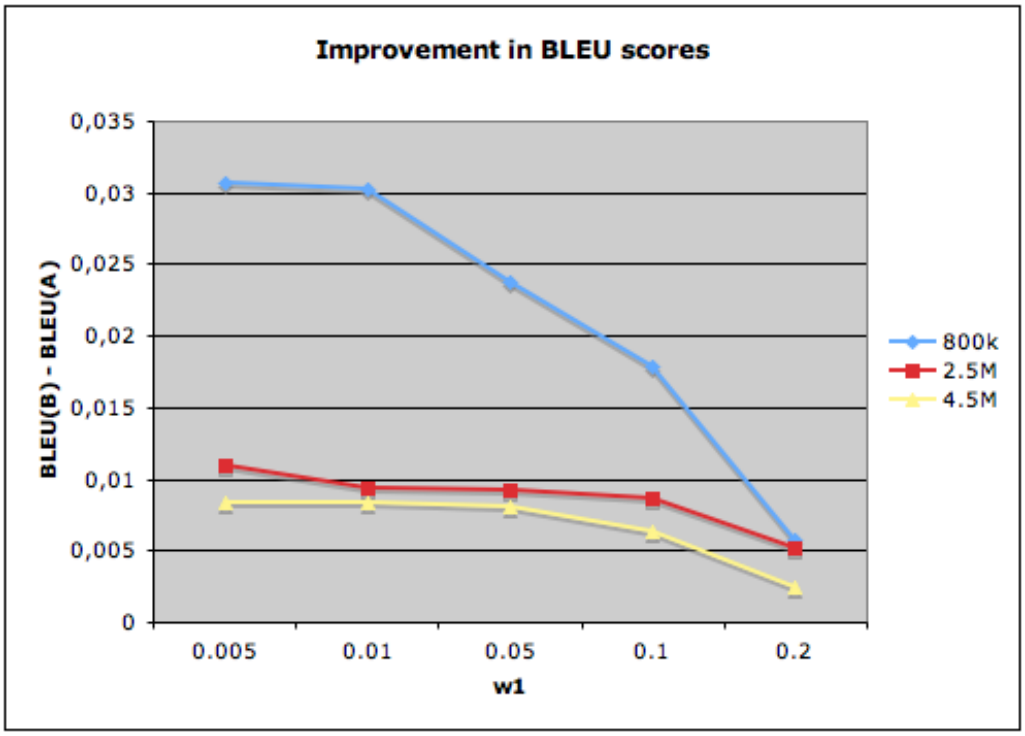

Figure 5.2: Improvement in BLEU scores (over w_1)

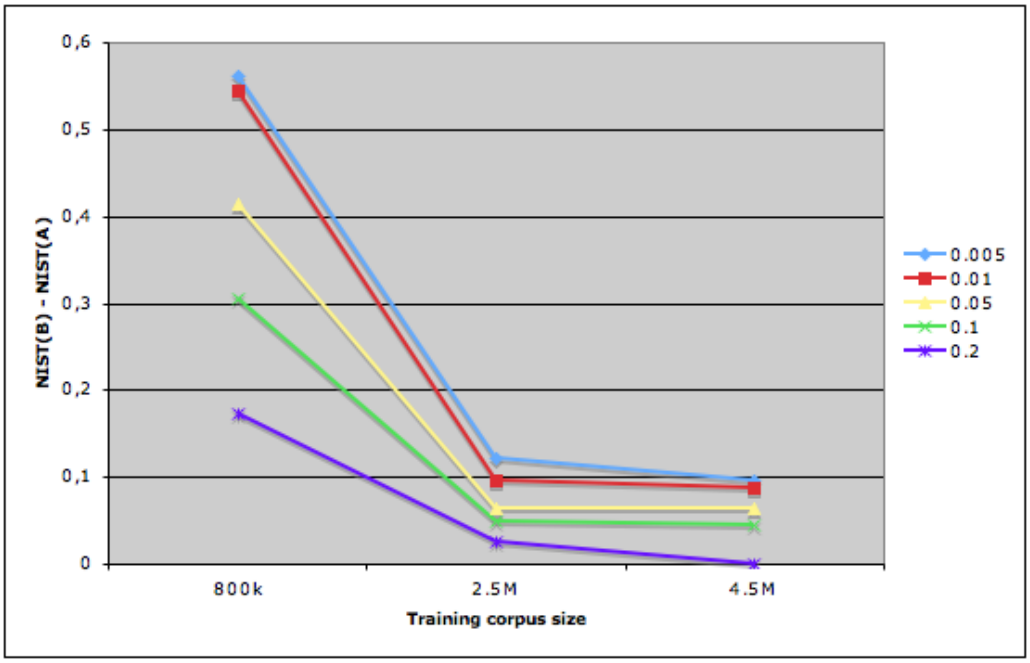

Figure 5.3: Improvement in NIST scores (over training set)

figures 5.3 and 5.4 show the improvement over the different training corpus sizes (x-axis). By examination of these figures it is evident that smaller values for w_1 result in more improvement in translation quality, as expected. The percentages of improvement for the two evaluation metrics, for the 800k, 2.5M and the 4.5M sets are presented in Tables 5.3, 5.4 and 5.5.

w_1	NIST improvement	BLEU improvement
0.005	14.72%	33.26%
0.01	14.14%	32.34%
0.05	10.39%	24.28%
0.1	7.40%	17.28%
0.2	4.06%	5.10%

Table 5.3: Percentage of improvement for the 800k set

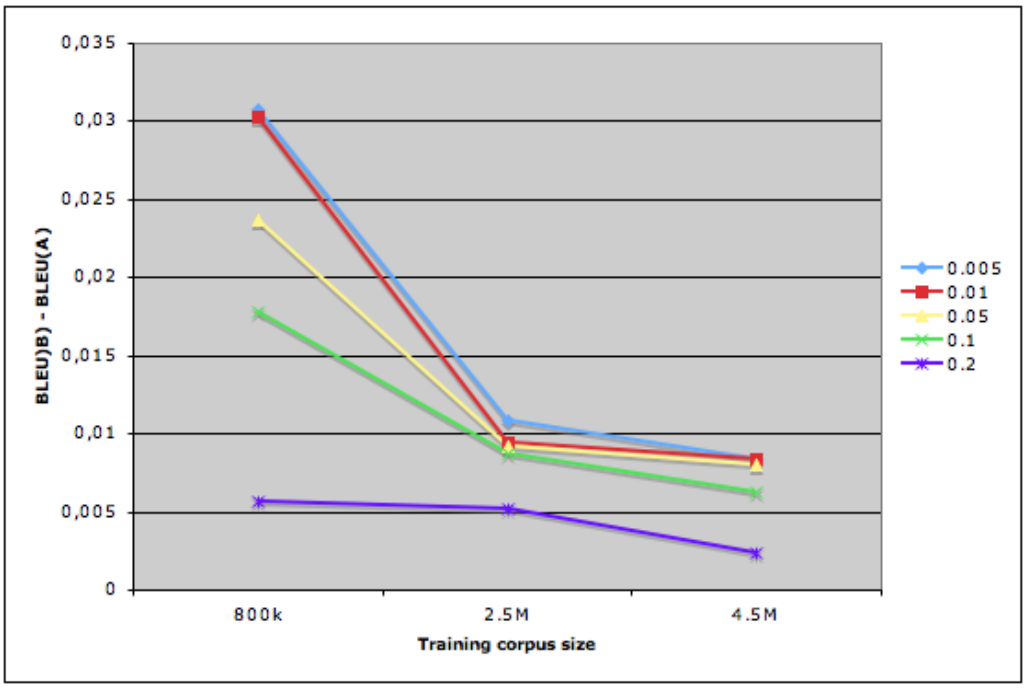

Figure 5.4: Improvement in BLEU scores (over training set)

w_1	NIST improvement	BLEU improvement
0.005	3.24%	8.18%
0.01	3.21%	7.49%
0.05	3.03%	7.01%
0.1	2.90%	6.23%
0.2	2.47%	3.51%

Table 5.4: Percentage of improvement for the 2.5M set

w_1	NIST improvement	BLEU improvement
0.005	2.28%	7.41%
0.01	2.28%	7.41%
0.05	2.25%	6.96%
0.1	1.97%	5.38%
0.2	1.36%	1.91%

Table 5.5: Percentage of improvement for the 4.5M set

5.7 Statistical Significance

5.7.1 Practical limitations

In statistical machine translation, the size of the test set is an important factor in the validity of a system's results. The bigger the test set, the more significant the results. However, it is extremely difficult to conduct experiments using huge test sets for two reasons. First, the procedure of translating takes a lot of time, especially when using large lattices for every sentence translated (in our case each lattice occupied several MBs on disk). Second, there is lack of such huge parallel corpora. In order for the automatic evaluation to take place, there must be a reference translation, to which the output of the systems tested is compared. Based on this reference, the automated evaluation procedure computes the values for the evaluation metrics. This forces the test sets to be taken from bilingual corpora, so in our case we had to split the corpora in three parts; one part for the training set, a second part for development and a third part for testing. This limits the available size of the test set.

In order to overcome these limitations and be certain that our results are valid, we performed statistical significance tests in our results. These tests were performed to validate our results, thus providing extensive evidence that the improvement that has been noticed in the combined lexical-morphological system is indeed true.

5.7.2 Confidence intervals

Statistical significance is an estimate of the degree, to which the true translation quality lies within a confidence interval around the measurement on the test sets. A common level or reliability that is used is that of 95%, or $p = 0.05$. The

interval represents the range of values, consistent with the data, which is believed to encompass the "true" value with high probability, in this case 95%. The confidence interval is expressed in the same units as the estimate. Wide intervals indicate lower precision; narrow intervals, greater precision. The estimated range is calculated from a given set of sample data.

5.7.3 Bootstrap resampling

Bootstrap resampling, or bootstrapping[2], is a data-based statistical method for statistical inference, which can be used to measure confidence intervals. It has a long tradition in the field of statistics [3].

The method of bootstrap resampling is based on the following assumption: estimating the confidence interval from a large number of test sets with n test sentences drawn from a set of n test sentences with replacement, is as good as estimating the confidence interval for test sets of size n from a large number of test sets with n test sentences drawn from an infinite set of test sentences.

In practice, the method is relatively simple. Having a set of n test sentences that we evaluated, we create a new set of the same number of sentences. This set is created from the initial set of sentences with replacement, that is for the first sentence of a new set, a sentence of the initial set is chosen randomly, then for the second etc. This means that in the new set, a sentence from the initial set may exist zero or many times.

Repeating the process explained above, we come up with a new set of n sentences. This is considered to be the second sample. In order to create more samples (sets) the process is repeated for the desired amount of samples. Let b be

[2]The term *bootstrapping* refers to the old story about people lifting themselves off the ground by pulling on the backs of their own boots.

the number of samples created. Every sample is evaluated, in our case resulting into scores for the BLEU and NIST values. As it is expected, these scores have a normal distribution. From these samples, the confidence interval is calculated by keeping the middle 95% scores (from the 2.5^{th} percentile to the $97,5^{th}$ percentile).

After calculating the confidence interval, the mean and relative standard deviation are calculated. *Relative standard deviation*, or RSD is defined as $(100 * \sigma/\mu)\%$, where μ and σ are the mean and standard deviation respectively.

This method has been used in various fields of research, including automatic speech recognition and statistical machine translation [24], [38], [39].

5.7.4 Statistical significance test results

Tables 5.6 and 5.7 display the 95% confidence intervals as well as the relative standard deviation that has been calculated for various experiments, for the NIST and BLEU metrics respectively. TC_s denotes the training corpus size, N_dmean the mean value found, N_d interval the confidence interval values and N_B RSD the relative standard deviation for the combined lexical-morphological system.

TC_s	w_1	N_d mean	N_d interval	N_B RSD
800k	0.005	0.5616	[0.4793, 0.6453]	1.04%
800k	0.05	0.4143	[0.3465, 0.4849]	0.99%
4.5M	0.005	0.0993	[0.0426, 0.1572]	1.14%
4.5M	0.05	0.0632	[0.0101, 0.1156]	0.99%

Table 5.6: 95% confidence intervals for N_d scores (NIST)

TC_s	w_1	B_d mean	B_d interval	B_B RSD
800k	0.005	0.0169	[0.0123, 0.0214]	2.33%
800k	0.05	0.0136	[0.0095, 0.0178	2.26%
4.5M	0.005	0.0054	[0.0016, 0.0094]	2.64%
4.5M	0.05	0.0035	[0.0002, 0,0071]	2.26%

Table 5.7: 95% confidence intervals for B_d scores (BLEU)

Chapter 6

Conclusions

In this work, we have presented a novel algorithm for the incorporation of morphological knowledge into existing statistical machine translation systems. Most SMT systems today operate at the lexical level without taking into account the morphologies of languages they're translating to and from. Morphological knowledge, such as an analysis of a word into it's stem and affixes, could be incorporated into these lexical SMT systems, clearly improving their performance, as our experiments have shown.

Using an unsupervised method for the morphological analysis of the texts to be translated, rules are obtained that specify the root and affixes that construct words. In the first phase, this information is used in order to stem the training corpora and use them to create a stem-to-stem SMT system. A mathematical framework that incorporates morphological knowledge has been presented, resulting in a morphological SMT system that performs word-level translation through the stem-level system. This system is then combined with a traditional lexical system, resulting in the combined lexical-morphological SMT system.

Such a system has been implemented using late integration and lattice rescoring. This new system has been compared to the baseline system, regarding the translation quality of their output. Evaluation has been performed by computing

automatic evaluation metrics that are widely used in MT, namely the BLEU and NIST metrics. The combined SMT system has proven to improve the translation quality up to 14% for the NIST metric and up to 33% for the BLEU metric for the sentences that result in different translations between the two systems.

In order to be certain that our results are valid and that the test set that have been used is large enough to provide confident results, we performed significance tests over our results. Using the method of bootstrap resampling, we calculated the confidence intervals of the two metrics. Examination of these confidence intervals has shown that our experimental results are statistically significant and that the new proposed SMT system has clearly improved on the baseline.

The combined lexical-morphological SMT system that has been implemented is using information about word stems, discarding affixes. In order to address the possible information loss that that may take place due to this, we propose in Section 6.1 an extension of the statistical framework that exploits affix information as well, by computing an affix-to-affix translation model and incorporating it into the SMT formulation.

Our work has shown that SMT systems can be enhanced greatly by incorporating morphological knowledge in them, using a framework like the one we propose. This performance boost is mostly evident in systems that are badly trained, because of the data in the training corpora being sparse.

As the field of statistical machine translation may be considered to be in it's infancy, it is evident that pure lexical SMT systems will soon be a thing of the past. Spoken languages are extremely complex constructs that are quite difficult to be modeled. Incorporating morphological, syntactic or semantic information into existing SMT systems shall definitely improve their performance.

6.1 Ongoing work

This section introduces the ongoing work that has been carried out in order to implement a system that exploit affix information and incorporates it into the morphological SMT system that has been presented in this work.

6.1.1 Affix-to-affix translation

In the same way that the word-level and stem-level SMT systems are created, it is possible to create an affix-to-affix SMT system. During the training stage, the morphological analyzer creates two output texts: one with the stemmed corpus, and one with only the affixes of the words of the original corpus. This corpus is then used to train the language and translation models that operate in the affix level. Given an affix A_s in the source language, the best translation of it $\hat{A}_t$ in the target language would be:

$$\hat{A}_t = \arg\max_{A_t} P(A_t|A_s) \tag{6.1.1}$$

6.1.2 Extended SMT system

In order to incorporate this affix-to-affix translation system into the system described so far, an alteration to the topology of the bayesian network is required. Word-to-word translation should now be achieved not only through the stem-to-stem system, but by using this affix SMT system as well. The new topology is depicted in Figure 6.1.

For a Bayesian network like this, we have:

$$\begin{aligned} \hat{W}_t &= \arg\max_{W_t} P(W_t|W_s) \\ &= \arg\max_{W_t} \sum_{S_s,S_t,A_s,A_t} P(W_t, S_s, S_t, A_s, A_t|W_s) \end{aligned} \tag{6.1.2}$$

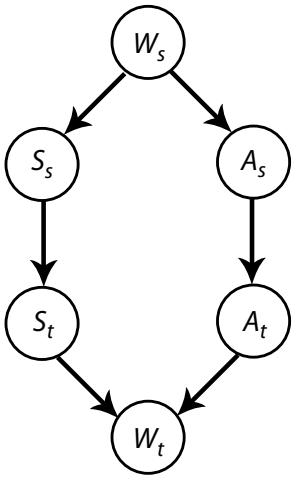

Figure 6.1: Enhanced Bayesian Network Topology

but assuming that W_t, W_s are conditionally independent given S_s, S_t, A_s, A_t; A_t, W_s are conditionally independent given A_s; S_t, W_s are conditionally independent given S_s, eq. (6.1.2) becomes:

$$\hat{W}_t = \arg\max_{W_t} P(W_t|W_s)$$

$$= \arg\max_{W_t} \sum_{S_s,S_t,A_s,A_t} P(W_t|S_t,A_t)P(S_t|S_s)P(S_s|W_s)P(A_t|A_s)P(A_s|W_s) \quad (6.1.3)$$

which can be further simplified since $P(S_s|W_s) = 1$ and $P(A_s|W_s) = 1$ since we assume that the mapping from a word to a stem and affix is deterministic, resulting in:

$$\hat{W}_t = \arg\max_{W_t} \sum_{S_t,A_t} P(W_t|S_t,A_t)P(S_t|S_s)P(A_t|A_s) \quad (6.1.4)$$

The probability model $P(W_t|S_t, A_t)$ is hard to compute though, but using Bayes rule it can be written as:

$$P(W_t|S_t,A_t) = \frac{P(S_t,A_t|W_t)P(W_t)}{P(S_t,A_t)} \quad (6.1.5)$$

but because of the assumptions mentioned before it can be simplified into

$$P(W_t|S_t, A_t) = \frac{P(S_t|W_t)P(A_t|W_t)P(W_t)}{P(S_t, A_t)} \tag{6.1.6}$$

Combining Eq. (6.1.6) with Eq. (6.1.4) we have:

$$\hat{W}_t = \arg\max_{W_t} \sum_{S_t, A_t} \frac{P(S_t|W_t)P(A_t|W_t)P(W_t)P(S_t|S_s)P(A_t|A_s)}{P(S_t, A_t)} \tag{6.1.7}$$

but since the maximization is done over both S_t and A_t the probability $P(S_t, A_t)$ is always equal to 1 so the probability of the denominator may be omitted.

6.1.3 Additional affix information incorporation

This section provides a different approach for the incorporation of affix information into the SMT system. The idea of the proposed method is based on the assumption that the training corpora could also be used to model the probability of a specific stem being applied to an arbitrary word, when it occurs after a series of $n-1$ other words, in the target language.

This can be made clearer with the use of an example. By examining the training corpora for the target language, we take sets of n words, for example $n = 3$. Then, for a given word in the corpus, the $n - 1 = 3 - 1 = 2$ words are examined in relation to the affix of the word, i.e. in the sentence *he is driving*, we see that the word *driving* is being preceded by the words *he* and *is*. We stem the word *driving* and keep only it's affix: *-ing*. Then, we correlate this affix with the words that precede it. By counting how many times this pattern appears in the training corpora, we can assign it a probability. Such probabilities can be calculated for every affix of the target language and the previous $n-1$ words, in a way of an n-gram.

Some examples of such probabilities could be:

$$P(\text{-ing}|\text{he is}) = 0.89$$

$$P(\text{-ed}|\text{he has}) = 0.74$$

$$P(\text{-ation}|\text{he is}) = 0.03$$

The first probability denotes that it is quite probable that the word which follows the phrase *he is* should have the affix *-ing*; the last probability shows that it is not so probable for the word that follows the phrase *he is* to have the suffix *-ation*.

Calculation based on stemmed corpora

Following the motivation of the original morphological SMT system, we can train these probabilities on the stemmed training corpora, since the data are less sparse in the stem level. This way, the system will be trained more efficiently. Such a probability would then be, the probability that any word that follows a series of specific $n-1$ stems, ends with a specific suffix.

Formulation

Let a denote an affix and $s_{n-1}, s_{n-2}, \cdots, s_1, s_0$ denote a series of stems. Then we can compute:

$$P(a|s_{n-1}, s_{n-2}, \cdots, s_1) \tag{6.1.8}$$

for every possible affix a and all the sequences of n words in the training corpora. Then, when searching for the best possible affix a for a given stem s_0 we have:

$$\hat{a} = \arg\max_a P(a|s_{n-1}, s_{n-2}, \cdots, s_1)$$

which means that the most possible affix is the one that maximizes the probability of it being used in a word that is preceded by the rest $n-1$ words. This way, when the morphological generator is asked to generate a word from a stem, it does so in a way that takes into consideration the words *before* it, estimating what should be the proper affix for the word that is the last of a series of n words. The value of n could be assigned to the value that maximizes the efficiency of the system (like the use of the proper value for n-grams in traditional SMT systems).

Modeling fluency

The morphological generator should also guarantee that the word it generates is a valid english word. For this, there has to be a way of giving higher probability for valid words to be created than invalid or rare ones. This can be modeled by computation of the probabilites $P(a|s)$, which is the probability of an affix a being used by a particular stem s.

For example, P(s|play) could have a high value, since the word *plays* is a valid word and may appear often in the corpus. However, the probability P(ness|play) could be zero, since the word *playness* does not exist in the english vocabulary.

Combining likelihood and fluency

Both of the models that have been discussed so far can be used by the morphological generator in the process of generating the most probable word from a stem. This incorporation could be formulated as:

$$\hat{a} = \arg\max_{a} P(a|s_{n-1}, \cdots, s_1)P(a|s_0) \tag{6.1.9}$$

where a is the affix, that maximizes the probability of i) being used in a word that is preceded by a series of specific stems and ii) in conjunction with the stem

s_0 results in a valid and probable word.

This affix could then be concatenated to the stem, producing the most probable word. Obviously, the morphological generator could be instructed to return not only the most probable word, but a list of words, each one assigned to a probability that is computed using equation 6.1.9.

6.2 Future work

In the future we are looking forward into conducting experiments with larger training sets, as well as between pairs of languages other than english and greek. Evaluation of a broader range of such experiments will result in fine-tuning our system and providing further evidence on the improvement that morphological information can provide in SMT systems. We are also looking forward into implementing the enhanced SMT system that has been proposed in order to exploit affix information and evaluate the further improvement that this incorporation will possibly have to the translation quality of the existing SMT system.

The implementation of the morphological analyzer in our work assumes that the process of stemming a word is deterministic, i.e. given a word there is only one stem that the word is derived from. This is not always the case, especially for words that are derived from different but similar stems, since the morphological analyzer may falsely stem them. Employing a probabilistic model for the morphological analyzer could slightly increase the output quality. Also, experimentation could be done in how the performance of the analyzer affects the final translation quality, by changing the values for the heuristics of minimum stem length and ratio for the process of stemming.

Our work can also be generalized to incorporate not only morphological, but

linguistic information in general, like syntactic or semantic information. The mathematical formulation that has been presented can be easily generalized to deal with linguistic tags, instead of just stems. These linguistic tags may be obtained by the analysis of the texts in the shallow syntactic level (e.g. part-of-speech tags), deep syntactic or semantic level. In analogy to the morphological analyzers and generators that have been presented in this work, it is possible to implement such analyzers and generators that operate in these levels and incorporate them into SMT systems. This would result in a combination of the traditional lexical SMT system, with the morphological SMT system we have built and the new SMT systems, like the part-of-speech (POS) SMT system, semantic SMT system etc. Such a combination should also be accompanied by experimentation on the weights that are used to model the confidence of each system.

Specifically, we are looking forward into incorporating part-of-speech knowledge into our system and evaluating the improvement it will possibly have on our existing combined lexical-morphological system. The implementation of a POS tagger, paired with slight modification of the code that has been produced in this research can result in a new SMT system that will incorporate morphology and part-of-speech knowledge into the baseline system, thus providing further improvement in translation quality.

Appendix A

Translation samples

This section presents some translation examples for both systems. First the input sentence is displayed, followed by the translation of system A (the baseline system) and that of system B (the combined lexical-morphological SMT system). Finally the reference translation is displayed.

800k training corpus, $w_0 = 0.05$

Input: *however the main concern remains*

System A: *ωστόσο το βασικό μέλημα ακόμη*

System B: *ωστόσο η μεγάλη ανησυχία παραμένει*

Reference: *παρόλα αυτά η βασική ανησυχία παραμένει*

Input: *therefore the responsibility of the european union has to be seen in a certain context*

System A: *συνεπώς η ευθύνη την ευρωπαϊκή ένα συγκεκριμένο περιεχόμενο της ευρωπαϊκής ένωσης*

System B: *συνεπώς η αρμοδιότητα της ευρωπαϊκής ένωσης πρέπει να θεωρηθεί σε*

ένα συγκεκριμένο περιεχόμενο

Reference: *αρα οι ευθύνες της ευρωπαϊκής ένωσης εντάσσονται σε συγκεκριμένο πλαίσιο*

Input: *my final point concerns the preparation of the budget for*

System *A*: *τελευταίο σημείο αφορά την προετοιμασία του προϋπολογισμού*

System *B*: *το τελευταίο σημείο αφορά την προετοιμασία του προϋπολογισμού*

Reference: *θα αναφερθώ τέλος στην κατάρτιση του προϋπολογισμού του*

Input: *it is easy to see that developing countries still stand to benefit*

System *A*: *είναι εύκολο να διαπιστώσουμε ότι αναπτυσσόμενων χωρών ακόμη υψώσει προς όφελος*

System *B*: *είναι εύκολο να διαπιστώσουμε ότι οι αναπτυσσόμενες χώρες ακόμη υψώσει προς όφελος*

Reference: *είναι προβλεπόμενο ότι εδώ θα παρουσιαστούν πλεονεκτήματα για τις χώρες αυτές*

Input: *after all this is not a product which is a threat to people' s health*

System *A*: *σε τελική ανάλυση πρόκειται ούτε ενός προϊόντος που συνιστά απειλή για λαϊκό υγεία*

System *B*: *σε τελική ανάλυση δεν πρόκειται για ένα προϊόν που συνιστά απειλή για λαϊκό υγεία*

Reference: *ούτως ή άλλως δεν πρόκειται για ένα προϊόν βλαβερό για την υγεία*

Input: *we do not believe that a declaration is sufficient*

System *A*: *είμαστε της άποψης ότι μια δήλωση είναι επαρκείς*

System B: *δεν πιστεύουμε ότι μια δήλωση είναι επαρκείς*

Reference: *θεωρούμε πως μια διακήρυξη δεν είναι αρκετή*

Input: *we have to expect that it will take time to implement this report*

System A: *θα πρέπει να περιμένουν που θα τη φορά να την έκθεση*

System B: *θα πρέπει να αναμένουμε ότι θα λάβει καιρός να εφαρμόσει την έκθεση*

Reference: *πρέπει να περιμένουμε ότι θα χρειαστεί χρόνος για την υλοποίηση της εν λόγω έκθεσης*

4.5M training corpus, $w_0 = 0.05$

Input: *let me just look at the figures from the day that the agreement was signed*

System A: *επιτρέψτε μου να το ημέρας ότι η συμφωνία αυτή υπογράφηκε*

System B: *θα ήθελα μόνο να εξετάσει τα στοιχεία από την ημέρα που η συμφωνία που υπογράφηκε*

Reference: *επιτρέψτε μου μια σύντομη αναδρομή στους αριθμούς από την ημέρα της υπογραφής της συμφωνίας*

Input: *that is why big industry is so much in favour of them*

System A: *γι΄ αυτό οι μεγάλες βιομηχανίες είναι τόσο ευνοϊκά τους*

System B: *γι΄ αυτό οι μεγάλες βιομηχανίες είναι τόσο υπέρ αυτών*

Reference: *γι΄ αυτό και οι μεγάλες βιομηχανίες είναι υπέρ της χρησιμοποίησης αυτών των ουσιών*

Input: *we do not wish the charter to be a mere declaration*

System A: *δεν θέλουμε ο χάρτης πρέπει να γίνει μια απλή διακήρυξη*

System B: *δεν θέλουμε να γίνει μια απλή διακήρυξη του χάρτη*

Reference: *θέλουμε ο χάρτης να μη συνιστά απλά μια διακήρυξη*

Input: *i would like to thank the commissioner for his reply which i find extremely satisfactory*

System A: *θα ήθελα να ευχαριστήσω τον επίτροπο για την απάντηση που θεωρώ ιδιαίτερα ικανοποιητικό*

System B: *θα ήθελα να ευχαριστήσω τον επίτροπο για την απάντηση που θεωρώ απολύτως ικανοποιητική*

Reference: *ευχαριστώ τον επίτροπο θεωρώ την απάντησή του ιδιαίτερα ικανοποιητική*

Input: *i therefore support this amendment*

System A: *υποστηρίζω την τροπολογία αυτή*

System B: *για το λόγο αυτό υποστηρίζω την τροπολογία αυτή*

Reference: *συνεπώς βλέπω ευνοϊκά αυτή την τροπολογία*

Bibliography

[1] *Srilm - the sri language modeling toolkit*, http://www.speech.sri.com/projects/srilm.

[2] R. Singh A. Ford and G. Martohardjono, *Pace panini*, 1997.

[3] B. Efron, and R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, 1993.

[4] Peter Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John Lafferty, Robert Mercer, and Paul Roossin, *A statistical approach to machine translation*, Computational Linguistics **16** (1990), no. 2, 79–85.

[5] Peter Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert Mercer, *The mathematics of statistical machine translation: parameter estimation*, Comput. Linguist. **19** (1993), no. 2, 263–311.

[6] K. Church and P. Hanks, *Word association norms, mutual information, and lexicography*, Proceedings of ACL 27, 1989, pp. 76–83.

[7] M. Creutz and K. Lagus, *Unsupervised discovery of morphemes*, Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02, 2002, pp. 21–30.

[8] Carl de Marcken, *Unsupervised language acquisition*, Ph.D. thesis, MIT, 1995.

[9] Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman, *Indexing by latent semantic analysis*, Journal of the American Society of Information Science **41** (1990), no. 6, 391–407.

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

[11] European Association for Machine Translation (EAMT), *What is Machine Translation?*, http://www.eamt.org/mt.html.

[12] Eric Gaussier, *Unsupervised learning of derivational morphology from inflectional lexicons*, Proceedings of the Workshop on Unsupervised Methods in Natural Language Processing, Association for Computational Linguistics, ACL'99 (Univ. of Maryland, SA.), 1999.

[13] John Goldsmith, *Unsupervised learning of the morphology of a natural language*, Computational Linguistics (2001), 153–189.

[14] Zellig Harris, *Structural linguistics*, The University of Chicago Press, 1951.

[15] W. J. Hutchins, *Machine translation: past, present, future*, John Wiley & Sons, Inc., New York, NY, USA, 1986.

[16] Bishop of Chester J. Wilkins, *An essay towards a real character and a philosophical language*, London, 1668.

[17] Christian Jacquemin, *Guessing morphology from terms and corpora*, Research and Development in Information Retrieval, 1997, pp. 156–165.

[18] K. Papineni and S. Roukos and T. Ward and W. Zhu, Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics.

[19] Panagiotis Karageorgakis, Alexandros Potamianos, and Ioannis Klasinas, *Towards incorporating language morphology into statistical machine translation systems*, IEEE Automatic Speech Recognition and Understanding Workshop, 2005.

[20] Dimitar Kazakov and Suresh Manandhar, *Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming*, Machine Learning (2001), no. 43, 121–162.

[21] Ioannis Klasinas, *Statistical machine translation incorporating morphlogical knowledge and using improved alignments*, Master's thesis, Technical University of Crete, Electronics & Computer Engineering Department, 2005.

[22] Kevin Knight, *Decoding complexity in word-replacement translation models*, Computational Linguistics **25** (1999), no. 4, 607–615.

[23] P. Koehn, *Pharaoh: a beam search decoder for phrase-based statistical machine translation models*, The 6th Conference of the Association for Machine Translation in the Americas (AMTA), 2004.

[24] M. Bisani and H. Ney, Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), may 2004.

[25] M. Mohri, F. C. N. Pereira, M. D. Riley, AT&T FSM Library ^{TM}- Finite-State Machine Library, http://www.research.att.com/sw/tools/fsm/.

[26] Harald Trost Marco Baroni, Johannes Matiasek, *Unsupervised discovery of morphologically related words based on orthographic and semantic similarity*, Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON) (Philadelphia), Association for Computational Linguistics, July 2002, pp. 48–57.

[27] B. Mickael, S. Murthy, and A. Lunsberg, *Discovering morphemic suffixes : A case study in mdl induction*, Fifth International Workshop on AI and Statistics (Ft. Lauderdale, florida), 1995.

[28] Fanouris Moraitis, *Σύστημα Αυτόματης Μετάφρασης χρησιμοποιώντας Στατιστικά Μοντέλα*, Diploma Thesis, Technical University of Crete, Electronics & Computer Engineering Department, 2004.

[29] S. Neuvel and S. Fulop, *Unsupervised learning of morphology without morphemes*, ACL Workshop Unsupervised Learning in Natural Language Processing, 2002.

[30] S. Nirenburg, *Knowledge and choices in machine translation, machine translation: Theoretical and methodological issues*, Cambridge University Press, 1987.

[31] F. J. Och and H. Ney, *Improved statistical alignment models*, October 2000, pp. 440–447.

[32] Franz Josef Och, *Ein beispielsbasierter und statistischer ansatz zum maschinellen lernen von natorlichsprachlicher ubersetzung*, Diploma Thesis, Universitat Erlangen-Nirnberg, Germany, 1998.

[33] P. Koehn, Europarl: A Multilingual Corpus for Evaluation of Machine Translation, Draft, Unpublished.

[34] K. Papineni, S. Roukos, T. Ward, and W. Zhu, *Bleu: A method for automatic evaluation of machine translation*, Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

[35] Patrick Schone and Daniel Jurafsky, *Knowledge-free induction of inflectional morphologies*, Proceedings of the North American Chapter of the Association of Computational Linguistics (NAACL-2001) (Pittsburgh, PA), 2001.

[36] William A. Gale & Kenneth W. Church, *A Program for Aligning Bilingual Corpora*, 1993, Bell Laboratories 600 Mountain Avenue Murray Hill, 07974.

[37] 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.

[38] Y. Zhang, S. Vogel, Measuring Confidence Intervals for the Machine Translation Evaluation Metrics, In: Proceedings of The 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2004), Baltimore, MD USA, oct 2004.

[39] Y. Zhang, S. Vogel, A. Waibel, Interpreting BLEU/NIST scores: How much improvement do we need to have a better system?, Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal, may 2004.

[40] D. Yarowsky and R. Wicentowski, *Minimally supervised morphological analysis by multimodal alignment*, Proceedings of the 38th Meeting of the Association for Computational Linguistics (Hong Kong) (K. Vijay-Shanker and Chang-Ning Huang, eds.), October 2000, pp. 207–216.