NOUN PHRASE TRANSLATION

by

Philipp Koehn

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

December 2003

Dedication

Meinem Vater gewidmet, dessen Vorbild mich inspirierte.

Acknowledgments

Research efforts that culminated in this PhD thesis were the main academic focus of the last six years of my life. There are many people who helped me stay focused and sane in pursuit of this goal.

I would like to thank my adviser, Kevin Knight, for giving my the freedom to explore many paths of research and giving me guidance along the way. My thesis work benefited also greatly from the other members of my committee, Daniel Marcu, Ed Hovy, Paul Rosenbloom and Daniel O'Leary.

The Information Sciences Institute has been a great place for me to develop myself as a researcher. I received a lot of inspiration, feedback, and insights that helped me with my work and broadened my horizons from fellow students and researchers, most notably Alexander Fraser, my long time office mate Kenji Yamada, Yaser Al-Onaizan, Irene Langkilde, David Pynadath, Deepak Ravichandran, Chin-Yew Lin, Chris Ackerman, Hal Daume, Dragos Munteanu, Radu Scoricut, Jay Modi, Sheila Tejada, Ion Muslea, Gal Kaminka, Jafar Adibi, Mike Junk, George Stephanopolis, and my compatriates Ulrich Germann, Ulf Hermjakob, and Franz Josef Och.

Researchers from other institutions also helped me gain better understanding of the research process: during my stay at AT&T Research Steve Abney, Michael Collins, and Julia Hirschberg and during my time at Whizbang Labs Dallan Quass and Chris Manning.

I would not have survived these years without the support, love, and friendship of my parents Johannes and Sigrid Köhn, my siblings Jan and Annette Köhn, my grandmother Traute Behr, Claudine Bastos Bayma, Christoph Steinebrunner, Michael Straeubig, Alexander Peters, and Pedja Klasnja in distant places, but also the good neighbors in Venice Beach: my girlfriend Traylove, James "Fly" Reynolds, Brian Gerkey, Matt Kuster, Samone, Warren "Malibu" Taft, Gianluca Colombo, Marwan Mograbe, Bill, Jimmy Wright, and Jenna Reynolds.

I would also like to than Dr. Jennifer Hearne, the dentist who pulled my four wisdom teeth, so I could finish writing the first full draft of this thesis under heavy pain medication. It has been a trip.

Contents

List Of Tables

List Of Figures

Abstract

We define noun phrase translation as a subtask of statistical machine translation. This enables us to build a dedicated noun phrase translation subsystem that improves over the currently best general statistical machine translation methods by incorporating special modeling and special features. We integrate such a system into a state-of-the-art statistical machine translation system with novel methods and show overall improvement in translation quality. We also carry out empirical linguistic studies on noun phrase translatability and the sources of translation errors.

Chapter 1

Introduction

1.1 Machine Translation

Natural Language Processing, the ability of computational systems to deal with human language in spoken or written form, is one of the core problems of Artificial Intelligence. It is no coincidence that the defining test for machine intelligence (the *Turing Test*) is in essence a natural language challenge: the ability to communicate with humans in their own language, and to be so *natural* in this ability that humans would not be able to tell the difference.

One of the main applications of Natural Language Processing is Machine Translation. Machine Translation is considered one of the *AI-hard* problems. For a system to be able to translate texts from one human language (say, Chinese) into another (say, English) much is required: knowledge about the corresponding meanings of words in the two languages, knowledge about the syntactic constraints of each language, semantic and pragmatic knowledge (world knowledge), and so on. These various forms of knowledge are necessary to resolve the ambiguities of natural languages that exist at various levels.

The successes of Statistical Machine Translation during the last decade underscores the importance of data-driven (or empirical) methods in this field. It seems that the vast amount of information necessary to guide the translation process can be most readily acquired from the vast amount of text that has already been generated by human translators. Data-driven methods automatically learn how to translate by analyzing such parallel corpora – texts along with their translations. The current approaches are relatively knowledge poor with respect to linguistic analysis: they rely only on automatically detected word correspondences and alignment patterns.

At the same time, we have seen a lot of progress in the construction of the infrastructure of basic linguistic tools for natural language processing: morphological analyzers, part-of-speech taggers, chunkers, parsers, ontologies, etc. These tools allow the automatic annotation of raw text with syntactic and even semantic information.

An ideal system for machine translation would take advantage of both empirical data and linguistic analysis. At this point, it is not clear how such a system should be constructed. Such a system has to address various distinct subtasks of the translation process: noun phrases, clause structure, discourse structure, anaphora resolution, and so on.

The work presented in this PhD thesis lays the foundations for such a system. We are narrowing the scope of investigation to noun phrase translation. The advantage of

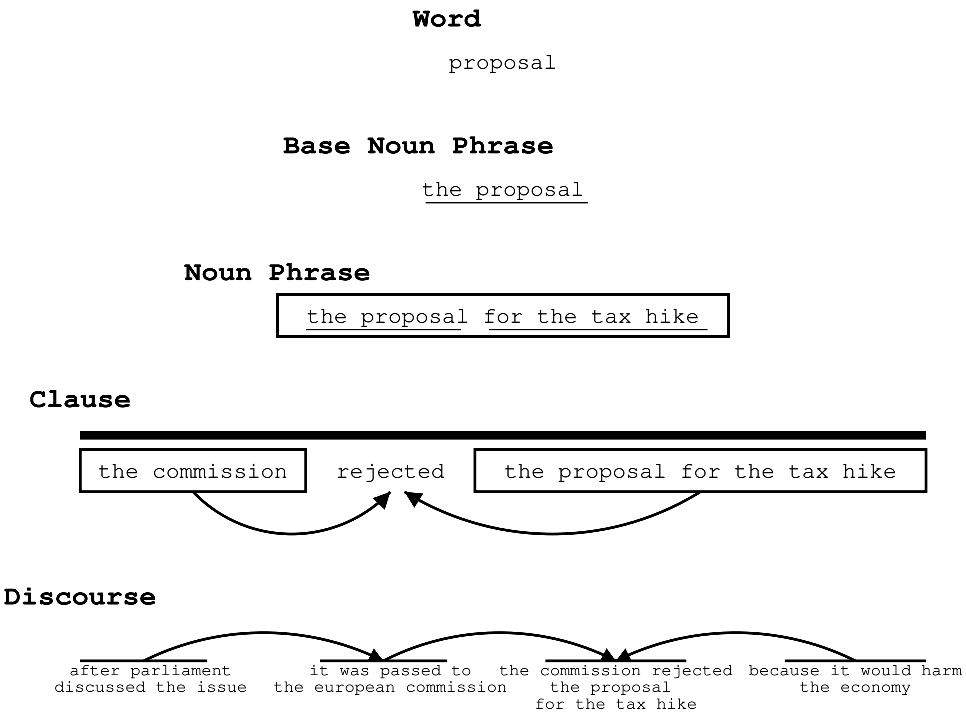


Figure 1.1: Five levels of syntactic structure: word, base noun phrase, noun phrase (the focus of this work), clause, and discourse

this focus is twofold: firstly, a more exhaustive study of the linguistic properties of this subtask can be carried out. Secondly, computationally more expensive methods can be applied. The results of this work should provide not only a better solution to the subtask of noun phrase translation, but also shine some light on the path to the envisioned ideal system for full machine translation.

1.2 Syntactical Structure

Figure 1.1 illustrates five levels of syntactic structure: word, base noun phrase, noun phrase, clause, and discourse.

Each of these levels poses challenges for translation. Different languages may differ in their syntactic structure in general: for instance the placement of the verb in clause structure or the use of prepositions or morphology to mark the role of base noun phrases. But also specific words and idiomatic expressions may force changes in syntactic structure.

Ultimately, a machine translation system has to take syntactic structure into account. Our general approach is to identify the characteristics of each of these levels and model them accordingly in a statistical machine translation system.

Some of our earlier work [Koehn and Knight, 2000, 2001, 2002b] addressed the word level: specifically, the acquisition of probabilistic lexical translation from various sources, such as bilingual dictionaries, monolingual and parallel corpora.

With this thesis, we move to the middle level of Figure 1.1: Noun phrases. Focusing on one syntactic category has a number of advantages. We can look more intensively at the characteristics of such a limited subproblem. Because the problem is smaller in size as well as complexity, we can apply computationally more expensive methods and focus on a more limited set of issues.

The lessons we learn from studying noun phrase translation will be useful when moving on to the next levels of syntactic structure. Noun phrase translation models will also become actual building blocks for models for those higher levels.

1.3 Definition of NP/PP

The scope of this work is the translation of noun phrases and prepositional phrases. As we define it, the **noun phrases and propositional phrases (NP/PP)** of a sentence are the maximal syntactic phrases that contain at least one noun and no verb.

We will first provide some examples to illustrate this concept and conclude this section with a formal definition. Consider the following opening sentence from a recent New York Times news item. The NP/PP are marked by italics and parentheses. Note that our definition includes not only baseNPs such as *the Bush administration*, but also more complex noun phrases such as *any involvement in a treaty for an international criminal court*.

NP/PP Example 1 *(The Bush administration)* has decided to renounce formally *(any involvement in a treaty for an international criminal court)* and is expected to declare that *(the signing of the document by the Clinton administration)* is no longer valid, *(government officials)* said today.

The definition of NP/PP excludes noun phrases that consist of only a pronoun. The major challenges for pronoun translation is verb subcategorization and anaphora resolution. We view these as separate from from the core task of noun phrase translation.

In the following example, the noun phrases *we* and *it* are not considered NP/PP, because they do not contain a noun.

NP/PP Example 2 "We think it was *(a mistake)* to have signed it," *(an administration official)* said.

Our definition also excludes noun phrases that contain relative clauses. Since the translation of relative clauses entails the same challenges as the translation of whole sentences, it would widen the subtask of noun phrase translation considerably.

In the following example, the prepositional phrase *a 1969 pact that outlines the obligations of nations* is not a NP/PP because it includes the verb *outlines*.

NP/PP Example 3 *(In addition)* , *(other officials)* said, *(the United States)* will simultaneously assert that it will not be bound *(by the Vienna Convention on the*

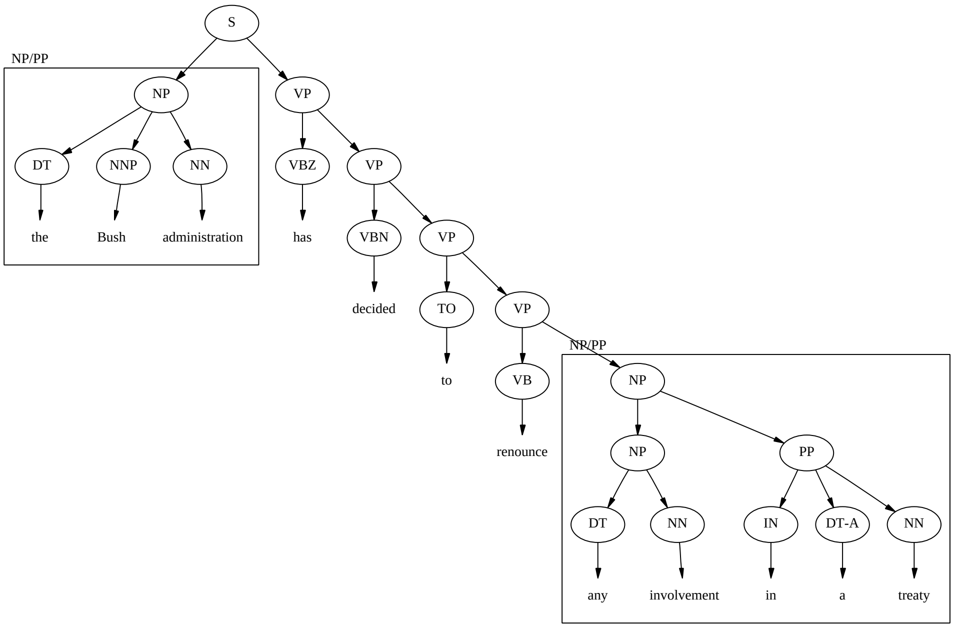


Figure 1.2: The NP/PP of this sentence are the framed boxes in the syntax tree: the maximal phrases that contain at least one noun and no verb

Law of Treaties) , *(a 1969 pact)* that outlines *(the obligations of nations)* to obey *(other international treaties)* .

NP/PP may contain connectives such as *and*, as the following example shows.

NP/PP Example 4 *(The permanent tribunal)* is designed to prosecute *(individuals) (for genocide, crimes against humanity and other war crimes)* .

Also note that two NP/PP may be adjacent to each other, if they do not have any direct syntactic relation: *individuals* and *for genocide, crimes against humanity and other war crimes* are both attached to the verb *prosecute*, and not to each other. Hence, they form two distinct NP/PP.

Let us conclude with a formal definition NP/PP in terms of syntactic structure.

Definition 1 Given a sentence s and its syntactic parse tree t, the NP/PP of the sentence s are the subtrees t_i that contain at least one noun and no verb, and are not part of a larger subtree that contains no verb.

This definition is illustrated by the parse tree in Figure 1.2. The boxed subtrees are NP/PPs.

Note that this definition includes related categories in different languages, such as *bunsetsu* in Japanese (noun phrases with case marking in post positions).

1.4 Characteristics of Noun Phrases

Having defined NP/PPs, let us take a closer look at their characteristics. It is helpful to distinguish the content of a noun phrase from the role the noun phrase plays in the context of the clause. In this work, we focus more on the content of the noun phrase, since the role is best dealt with at the clause level.

1.4.1 Content

From a semantic point of view, noun phrases describe objects or abstract concepts. This content is captured by one or more nouns and adjectives.

The vast majority of words in a language are nouns. As new objects in the world have to be named, new nouns are introduced. This poses a challenge for machine translation, since knowledge about these new nouns and their translation has be entered into the system. Acquiring this knowledge from parallel corpora (texts along with their translation) appears to be an elegant solution to this problem.

Noun phrases that consist of multiple nouns and adjectives may describe concepts in the world that can not be easily derived from these single nouns. Consider, for example, the noun phrase *interest rate*. Its meaning does not follow straight-forward from the highly ambiguous *interest* and *rate*.

Recent methods in statistical machine translation are based on the translation of word sequences (or phrases). This helps to overcome this problem by enabling the acquisition of translation entries for multi-word units such as *interest rate*.

Still, since the composition of concepts is a generative process, and new noun phrases are constantly formed, a pure memorization approach does not in fact suffice to solve the NP/PP translation problem.

1.4.2 Role

Noun Phrases play a certain syntactic roles at the clause level which indicates their relation to the verb: subject, object, adjunct, etc.

English strongly enforces a particular word order. The typical English sentence is of the form subject-verb-object (SVO). Hence, position indicates the syntactic role.

In addition to this, prepositions may be used to specify relations of noun phrases to the verb, e.g., *after 3 p.m.* or *in Los Angeles*. One-word markers similar to such prepositions may be required in other languages to indicate syntactic roles. For instance, the Japanese *ga* indicates the subject.

Languages may also use morphological variation to indicate the role of a noun phrase in the clause. For instance, in German, morphological changes to determiners, adjectives, and nouns indicate the syntactic case of a noun phrase and hence its role in the sentence.

All these – position, prepositions, and morphological variations – can function as markers that indicate the syntactic role of noun phrases. Languages vary widely in the choice of these markers. Fixed word order languages such as English rely on position and require little additional markup. More free word order languages such as German require a much richer morphology of determiners, adjectives and nouns. Markers may occur as

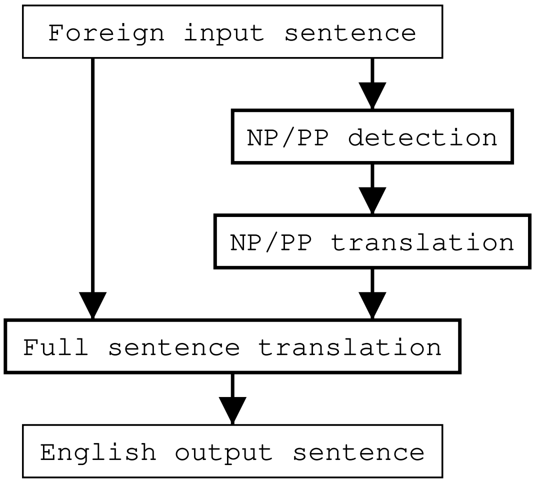


Figure 1.3: Divide and Conquer: NP/PPs of the input sentence are detected and translated by a separate NP/PP translation subsystem. The full sentence translation system integrates the NP/PP translations with the translations of the rest of the sentence.

separate words (determiners, prepositions), or, as in the case of languages such as Finnish and Arabic, include this role information as part of the morphology.

1.5 A Divide and Conquer Approach

We will now give an overview of our approach of noun phrase translation. We tackle noun phrase translation in a *divide and conquer* fashion: we detect the noun phrases of a sentence, translate them with a separate NP/PP translation subsystem, and then integrate them into a full sentence translation system that translates the rest of the sentence. Figure 1.3 illustrates this approach.

Breaking up the translation task into subtasks is generally good engineering practice for vast problems such as machine translation. Breaking out NP/PP translation allows us to build models and systems that are more dedicated to this subtask than a general method.

One might object that (1) not all NP/PPs translate as NP/PPs into another language so that a acceptable translation of the sentence can be formed and that (2) the translation NP/PPs requires the knowledge of external context. We will address these two concerns in the following two sections that report on empirical linguistic studies.

1.6 Translation of NP/PPs

To address the above potential objections, we carried out a study to examine what NP/PPs translate to in a typical parallel corpus. Clearly, we cannot simply expect that certain syntactic types in one language translate to equivalent types in another language. Equivalent types might not even exist.

In this preliminary study we wanted to answer the following questions:

- Do human translators translate noun phrases in foreign texts into noun phrases in English?
- When all noun phrases in a foreign text are translated into noun phrases in English, is an acceptable translation of the text possible?
- What are the properties of noun phrases that cannot be translated as noun phrases without rendering the overall sentence translation unacceptable?

We collected a parallel corpus from the proceedings of the European Parliament, which are accessible on the web. The proceedings are published in the eleven official languages of the EU. From these, we selected a sample of German and English texts and collected from these 100 German-English sentence pairs of medium length (12 words).

We consider a translation task from German to English. We marked the NP/PPs on the German side of the parallel corpus manually. This yielded 168 noun phrases and prepositional phrases according to our definition.

We examined if these units are realized as noun phrases in the English side of the parallel corpus. This is the case for 122 of the 168 NP/PPs.

Secondly, we tried to construct translations of these NP/PPs that take the form of NP/PPs in English within some overall acceptable translation of the sentence. We could do this for 164 of the 168 NP/PPs.

The four exceptions are:

- *(in Anspruch)* genommen — Gloss: *(in demand)* take
- *(Abschied)* nehmen — Gloss: *(good-bye) take*
- *(meine Zustimmung)* geben — Gloss: *(my agreement) give*
- *(in der Hauptsache)* — Gloss: *(in the main-thing)*

The first three cases are noun phrases or prepositional phrases that merge with the verb. This is similar to the English construction *make an observation*, which translates best into some languages as a verb equivalent to *observe*. The fourth example, literally translated as *in the main thing*, is best translated as *mainly*.

Why is there such a considerable discrepancy between the number of noun phrases that *can* be translated as noun phrases into English (98%) and noun phrases that *are* translated as noun phrases (75%)?

One caveat is that the chosen parallel sample text is mostly not generated in the form of a German to English translation process. Both the German and the English are often translations of an original in a third language. Also, sometimes the translations are sloppy.

A more significant reason is that translators generally try to translate the meaning of a sentence, and do not feel bound to preserve the same sentence structure. This leads them to sometimes restructure the sentence, in ways that often seem arbitrary.

The conclusion of this study is: most NP/PPs in German are translated to English as NP/PPs. Nearly all of them, 98 percent, can be translated in such a way into English,

Performance	Ratio
Translated correctly	89%
Wrong phrase start	9%
Wrong content word meaning	2%

Table 1.1: Human translation performance when translating NP/PPs without external context

while allowing satisfactory full sentence translation. The exceptions to this rule should be treated as special cases and handled separately.

We carried out preliminary studies for Chinese and Portuguese NP/PPs with similar results.

1.7 The Influence of External Context

One interesting question is whether external context is necessary for the translation of noun phrases. While the foreign sentence and document context may be available to the NP/PP subsystem, the English output context is only assembled later and therefore harder to integrate.

To address this issue, we carried out a manual experiment to check if humans can translate NP/PPs without any external context. Using the same corpus of 164 translatable NP/PPs as in the previous section, we asked human translators to translate the NP/PPs in isolation, without giving them the context in which they occurred.

The results are displayed in Table 1.1. A human translator translated 89% of the noun phrases correctly, 9% had the wrong leading preposition, and only 2% were mistranslated with wrong content words.

Picking the right phrase start (e.g., preposition or determiner) can sometimes only be resolved when the English verb is chosen and its subcategorization is known. Otherwise, sentence context does not play a big role: word choice can almost always be resolved within the internal context of the noun phrase.

To illustrate the difficulties if picking the right phrase start, consider the following two sentences:

- Ich ziele *(auf den Mann)* .
- Ich gehe *(auf den Mann)* zu .

These two sentences translate into English as:

- I aim *(at the man)* .
- I walk *(to the man)* .

The correct translation of the prepositional phrase depends on the sentence context, specifically the subcategorization of the English verb. We already noted in the previous section that such external knowledge may be important for the translation of NP/PP.

However, the subcategorization of the English verb is not available to the system before the English verb is chosen. For instance, the second sentence could also be translated as

- I approach *(the man)* .

Hence, the correct translation of the prepositional phrase may depend not only on the German context (which is available before sentence translation), but the also on the English context (which is only available afterward). The same observation holds for case marking when translating into a language with case.

1.8 Integration into an MT System

The findings of the previous sections indicate that a subsystem for NP/PP to NP/PP can be conceived of as a separate subsystem of a complete machine translation system – with due attention to special cases. We will now estimate the importance of such a system.

As a general observation, we note that NP/PPs cover roughly half of the words in news or similar texts. All nouns are covered by NP/PPs. Nouns are the biggest group of open class words, in terms of the number of distinct words. New nouns are constantly added to the vocabulary of a language, be it by borrowing foreign words such as *Fahrvergnügen* or *Zeitgeist*, by creating new words from acronyms such as *AIDS*, or by other means. In addition to new words, new phrases with distinct meaning are constantly formed: *web server*, *home page*, *instant messaging*, etc. Learning new concepts from text sources when they become available seems to be an elegant solution for this knowledge acquisition problem.

In a preliminary study, we assessed the impact of a NP/PP subsystem on the quality of an overall machine translation system. We tried to the answer the following questions:

- What is the impact on a machine translation system if noun phrases are translated in isolation?

- What is the performance gain for a machine translation system if a NP/PP subsystem provides perfect translations of the noun phrases?

First, we forced a state of the art word-based statistical machine translation system [Germann, 2003] to translate the NP/PPs in isolation. This meant that the NP/PPs were translated as a contiguous unit. No words could be moved out of a NP/PP and no words could be moved in. However, word choice within the NP/PPs and outside might be influenced by neighboring words.

Second, we built a subsystem for NP/PP translation that used the same modeling as the overall system, but is trained only on NP/PPs. With this system, we translated the NP/PP in isolation, without the assistance of sentence context. These translations were

System	Sentences Correct
Basic MT system	7%
NP/PP translated in isolation	8%
Perfect NP/PP	24%

Table 1.2: Performance when integrating a NP/PP subsystem into a full sentence translation system

fixed and provided to the general machine translation system, which could not change the fixed NP/PP translation.

Third, we provided correct translations for the NP/PP to the general machine translation system.

We carried out these experiments on the same 100 sentence corpus described in the previous section. The 164 translatable NP/PPs are marked and translated in isolation.

The results are summarized in Table 1.2. Treating NP/PP as isolated units and translating them in isolation with the same methods as the overall system had little impact on overall translation quality. A perfect NP/PP subsystem would triple the number of correct sentences.

These findings indicate that solving the NP/PP translation problem would be a significant step toward improving overall translation quality, even if the overall system is not changed in any way. The findings also indicate that isolating the NP/PP translation task as a subtask does not harm the overall translation system.

1.9 Existing Solutions

Since noun phrase translation has not been previously defined as a separate problem for statistical machine translation, there are no existing solutions tailored especially for noun phrase translation.

However, general statistical machine translation systems can be easily applied to noun phrase translation. For these systems, we need to collect a parallel corpus of NP/PPs, instead of whole sentences. Then, we simply train the systems on this data.

In a preliminary experiment we compared four different approaches:

- Word-Based (IBM Model 4)
- Chunk-Based (ChunkMT)
- Phrase-Based
- Memory-Based

The systems are described in more detail in the following chapter on related work (Chapter 2). IBM Model 4 is the result of seminal work by the IBM Candide group in the early 1990s [Brown et al., 1990]. ChunkMT is a statistical machine translation system that takes advantage of part-of-speech tags and syntactic chunks, proposed in our earlier

System	Coverage	Accuracy
Memory-Based	72	54
Model 4	164	83
ChunkMT	155	90
Phrase-Based	164	99

Table 1.3: Performance of a number of existing systems on NP/PP translation: Not all systems could translate all NP/PPs (coverage less than 164)

work [Koehn and Knight, 2002a], similar to an approach by Schafer and Yarowski [2003]. The phrasal translation system considered here is provided by Marcu and Wong [2002]. The fourth approach is simple memorization of all the noun phrases. If the noun phrase to be translated has been observed in the training data, the most frequent translation for it found in the training data is given as output.

To train these systems, we needed a parallel corpus of NP/PPs. We collected this corpus as follows: Given the 350,000 sentence Europarl German-English parallel corpus of sentences, we aligned the words in the corpus with the Giza toolkit [Al-Onaizan et al., 1999] and detected corresponding syntactic chunks with the ChunkMT training system. We also parsed the English side of the corpus with the statistical syntactic parser by Collins [1997] .

Given the English parses for the sentences, we could mark NP/PPs on the English side according to our definition. For these marked NP/PPs, we used the detected chunk alignments to find corresponding German baseNPs and basePPs. If an English NP/PP was aligned to a verb in German, it was discarded. The remaining NP/PPs and the corresponding German words are extracted as a training corpus of 650,000 parallel NP/PPs pairs. Upon brief examination, the alignment quality is roughly 90%. The remaining 10% are misaligned phrases, often with one side having additional material. A more refined method to create such a corpus is presented in Section 3.2.

All the systems for this experiment can be trained on such a corpus, and can be used on test data in conjunction with a language model that was trained on the English side of this parallel NP/PP corpus.

We tested the performance of the systems on the 164 test NP/PPs described in the previous section. The results are summarized in Table 1.3.

The systems have different advantages: phrase-based translation is better at capturing some non-literal translations, while the ChunkMT system produces syntactically more well-formed output.

1.10 Unresolved Problems

The traditional statistical approaches, which we will describe in more detail later, share the following strategy: words are treated as tokens with no further linguistic markup. A training method tries to find word or phrase mappings from parallel sentences. Based on these alignments we can estimate word or phrase translation probability distributions, word movement probability distributions or alignment patterns, and probability distributions to add and drop words.

Generally speaking, translation is easy when all that has to be done is word-by-word translation with unambiguous dictionary lookup. Difficulties emerge when words have to be reordered, word translation is ambiguous, or a literal translation of certain word or phrases is not possible. A good machine translation system has to detect these cases and treat them properly.

We will now give a few examples to demonstrate the shortcomings of the existing solutions. These examples are taken from the 164 NP/PPs used in the experiment of the previous section.

Translation Example 1 Different syntactic structure

- **German:** *meiner Auffassung nach*
- **Gloss:** *my view after*
- **Correct Translation:** *in my view*
- **Model4 Translation:** *my view*

Grammatical structure may differ between source and target language. Simple examples for this are noun-adjective constructions in Romanic languages, or, in this example, a preposition that follows a noun phrase. Here, the challenge to the system is to move the preposition in front of the base noun phrase. Instead, in this case, the system decides to drop the preposition.

Translation Example 2 Ungrammatical output

- **German:** *mit den Geldern der Steuerzahler*
- **Gloss:** *with the money of-the taxpayers*
- **Correct Translation:** *with the taxpayers money*
- **Model4 Translation:** *with the money the taxpayer*

The second base noun phrase (*der Steuerzahler*) is a genitive construction, which may either turned into a prepositional phrase (*of the taxpayers*), or moved forward. However, current systems are not sensitive to such case marking. Moreover, the output in this example is ungrammatical, which is not addressed by n-gram language models that are not taking syntactic structure into account.

Translation Example 3 Phrasal translation

- **German:** *viele Kolleginnen und Kollegen*
- **Gloss:** *many colleagues and colleagues*

- **Correct Translation:** *many colleagues*
- **Model4 Translation:** *many members and colleagues*

In this example, the German phrase addresses both male and female colleagues. However, the English translation is not gender specific. The system produces two synonymous translations for the two nouns in its confusing, if not wrong, output.

Translation Example 4 Compounding

- **German:** *dem blutigen Vernichtungskrieg gegen das tschetschenische Volk*
- **Gloss:** *the bloody war-of-extermination against the Chechen people*
- **Correct Translation:** *the bloody war of extermination against the Chechen people*
- **Model4 Translation:** *the bloody Vernichtungskrieg against the Chechen people*

Compound words such as *Vernichtungs/krieg* occur frequently in German. In the worst case, the compounding of words creates new words that have not been seen in training, as is the case here. A translation system that is not sensitive to such compounding is left with the option to simply repeat the word verbatim, which only makes sense for proper names.

1.11 Overview

In this introduction, we motivated and defined the problem of noun phrase translation. Aided by a number of preliminary studies and experiments, we illustrated the characteristics of the problem and the shortcomings of existing solutions.

We showed that – in the language pairs we examined – noun phrases can almost always be translated into noun phrases, while still allowing for an acceptable fluent output sentence.

We will now give an overview of our work and how we present it in this thesis. The next chapter reviews related work in the field of machine translation in more detail. Chapter 3 to Chapter 5 describe the main body of this work. We close with Chapter 6, which reviews our contributions and conclusions. The appendix contains some elaboration on statistical significance and additional experimental results on the phrase-based translation model.

1.11.1 NP/PP Translation Module

We treat NP/PP translation as a subtask of machine translation. That means that we build a specialized translation module that translates one NP/PP at a time.

The approach we take in designing this module is a reranking approach: a base system (Section 3.3) provides an n-best list of candidate translations (Section 3.4). The base system is a phrase-based translation system that is trained on a NP/PP corpus (Section 3.2).

We enrich the candidate translations in each n-best list with additional features that may utilize additional knowledge sources. We use maximum entropy as the machine learning method that integrates the different features to rerank the n-best list (Section 3.5).

In our experiments, we can find an acceptable translation for 90% of all noun phrases in the n-best list, which justifies the reranking approach as a practical solution for noun phrase translation. The cause of error for the remaining 10% are mainly unknown words and failures of the tagging and parsing tools.

1.11.2 Properties of NP/PP Translation

In building our NP/PP translation model, we exploit a number of properties of NP/PP translation:

- compound splitting (Section 4.1)
- web n-grams (Section 4.2)
- syntactic features (Section 4.3)

We evaluate the performance of our system quantitatively in terms of performance (Section 4.4) as well as qualitatively in terms of remaining sources of error (Section 4.5).

In our experiments, adding modeling for each of these properties improves the translation performance. Overall, we can improve noun phrase translation accuracy from a baseline of 53.9% to 67.1%.

The main sources of reranking errors are the inability to predict the phrase start without context, wrong word choice for the translation of ambiguous words and phrases, and the dropping of content words.

1.11.3 Integration

We integrate the NP/PP translation module into two different full-sentence translation systems – word-based (Section 5.2) and phrase-based (Section 5.3). We develop methods to accomplish this integration, either by passing on the best translation chosen by the NP/PP translation module or by passing on a probability distribution over a set of translations (Section 5.4).

Integration into a word-based translation system leads easily to improvements of overall translation quality. Scored with the BLEU metric we improved performance from 0.176 to 0.198.

However, the integration of the noun phrase translation subsystem into a phrase-based translation system faces a number of difficulties: (1) Cutting out noun phrases eliminates the use of entries in the phrase-translation table that cross noun phrase boundaries. (2) There is loss of training data during the acquisition of a parallel corpus of noun phrases. (3) Tagging and parsing errors that lead to mistakes in identifying noun phrases may lead to failures of the noun phrase translation component.

Nevertheless, we can improve overall sentence translation performance of a phrase-based translation system with our noun phrase translation module by +0.003 on the BLEU metric. Key to this improvement is the flexibility of the integration: We pass the entire reranked n-best to the full system, and allow it to override the noun phrase module with its own translations.

1.12 Outlook

We successfully identified noun phrases as a syntactic category that defines a subproblem in statistical machine translation. Special modeling of its properties was exploited for improvement of the translation quality of noun phrases and, consequently, full sentences.

We introduced a number of techniques to deal with such a subproblem. Most notably the use of parsers to collect specialized training material from parallel corpora, the use of maximum entropy reranking to incorporate various features such as the very large corpus statistics and syntactic features, and a flexible integration framework to incorporate submodules into a full sentence translation system.

It was important to study the specific properties of the task at hand. By identifying the mistakes of the general-purpose phrase-based translation method, we were able to devise specific modeling techniques to address them. It was also important to recognize that the translation of natural language does not also break along linguistic categories, and therefore we need to provide a flexible framework that allows for exceptions.

At the beginning of this introduction, we outlined five levels of syntactic structure: word, base noun phrase, noun phrase, clause, and discourse. Breaking up the problem along these syntactic lines allows for the inclusion of syntactic knowledge at each level. Having addressed noun phrases, future research may build on this and address the next levels: clause and discourse structure. The lessons learned from our work on noun phrase translation should provide some guidance.

Chapter 2

Related Work

This is really written in English, but
it has been coded in some strange symbols.
I will now proceed to decode.
Warren Weaver, 1947

The development of machine translation systems can be traced back to the late 1940s. The great successes in breaking the encryption codes with computers during World War II led to the idea that the problem of machine translation could be solved in a similar fashion. This sparked a significant amount of research and interest by the early 1950s.

However, the failure to reach the lofty goal of fully automated high quality translation led to a collapse in funding and research. Machine translation was revived as a research topic in the late 1970s. Many research projects in the late 1970s and 1980s led to commercial ventures. Machine translation is now established as a research field and as a commercially viable application [Arnold et al., 1994].

2.1 Approaches to Machine Translation

We will first review the major directions of research in machine translation: interlingua, transfer-based, example-based and statistical.

2.1.1 Interlingua

Human translators read foreign words, and guided by their knowledge of foreign syntax and semantics form an understanding of the text that allows them to rewrite it in English aided by their knowledge of English semantics and syntax. This view of translation is symbolized by the machine translation pyramid in Figure 2.1.

Efforts to build machine translation systems that follow this model of the translation process face challenges: the acquisition of syntactic and semantic knowledge to transform foreign text into meaning representation on the one hand, and the generation of English text from meaning representations on the other hand. But above all, the representation of meaning in the form of an interlingua that is truly beyond language is really the hardest problem, since our understanding of meaning representation is very limited.

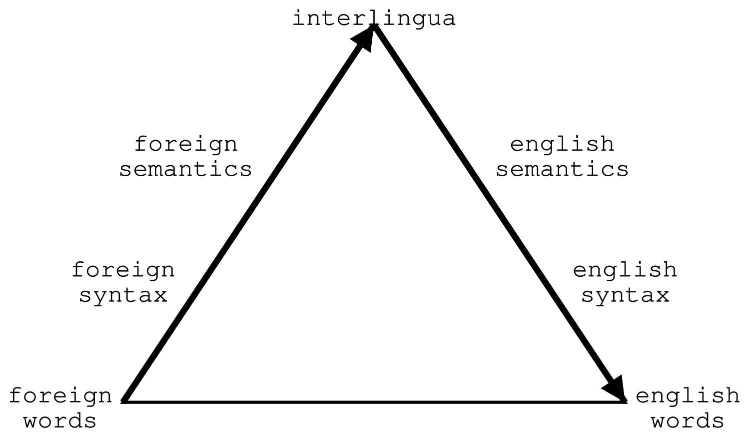


Figure 2.1: The machine translation pyramid

Still, this did not stop attempts to build interlingua-based machine translation systems. These approaches are also called knowledge-based, since they require a vast amount of knowledge resources (lexicons, grammar rules, and especially world knowledge) to transform words into meaning representations.

As of today, successful interlingua or knowledge-based machine translation systems are limited to small domains where it is feasible to assemble the required knowledge. However, the challenge to scale such systems to larger domains (say, news text) is one motivation behind various research efforts to build up such knowledge resources.

A more detailed description of knowledge-based machine translation is given by Nirenburg et al. [1992] and Arnold et al. [1994]. See also the description of the KANT system by III and Mitamura [1992], which is an implementation of this approach.

2.1.2 Transfer-Based

Transfer-based machine translation methods are related to knowledge-based interlingua methods in the sense that they also try to climb up the machine translation pyramid, but in contrast, not all the way to the top. The transfer from foreign structure to English structure takes place at some level below, ranging from limited syntax to some form of semantic representation.

The rules to create foreign structure, the transfer rules (lexical and structural), and the generation rules are usually hand-crafted. This requires some knowledge of *comparative grammar* of the language pair, i.e., what are the grammatical differences between the two languages. It is generally hard to resolve ambiguities in such systems, since there is no natural way to assign scores to the various rules.

A number of transfer-based machine translation systems are reviewed by Hutchins and Somers [1992]. Some of the concerns expressed about interlingua approaches are also valid here: the acquisition of grammar, transfer and generation rules is a sheer endless process, so that high-quality machine translation has been only achieved for limited domains.

2.1.3 Example-Based

Researchers that follow the interlingua and transfer-based approaches may consult a reference corpus of translated text as source of inspiration or validation and build their systems on the basis of such analysis and their own intuition. Contrast this to the empirical, or data-driven approach of example-based machine translation: Here, the machine learns to translate directly from a parallel corpus (text along with its translation). In its simplest form, a given input sentence is compared to a collection of sentences for which translations are known. The closest match is used to construct the output translation.

Various example-based machine translation methods differ in their matching criteria for "closest match", the length of input text that is being matched (sentences, or shorter fragments), the generalization of the stored translation examples, the degree of linguistic knowledge that is used during matching and generalization, etc. A good overview to example-based machine translation is presented by Somers [1999].

2.1.4 Statistical

Data-driven methods attempt to overcome the main problem of the more traditional symbolic approach: The need for a large human effort of linguistic analysis and rule writing is eliminated by the automatic acquisition of translation knowledge from a parallel corpus.

Statistical machine translation may be viewed as example-based machine translation with probabilities. However, historically it can be better understood as the continuation of methods that were highly successful in speech recognition: the decomposition of the problem into a generative statistical model.

Such a model is typically decomposed into a word to word (or phrase to phrase) translation model, a reordering model, and a language model. The model is trained to best explain the empirical data, i.e., the parallel corpus. The model is also used as scoring mechanism for possible translations for a foreign input sentence.

Our work falls into the statistical category of machine translation approaches. Therefore, we will describe the main statistical machine translation methods in more detail in the next section.

2.2 Statistical Machine Translation Methods

In the next sections, we describe related research in the field of statistical machine translation. We trace the progress from word-based models to phrase-based models and models that use syntactic transformation mechanisms.

2.2.1 Word Based: IBM Model 1-5

The Candide project at IBM [Brown et al., 1990] introduced many concepts for statistical machine translation that other researchers followed, such as the expectation maximization (EM) approach for training from parallel corpora and the noisy channel approach for decoding.

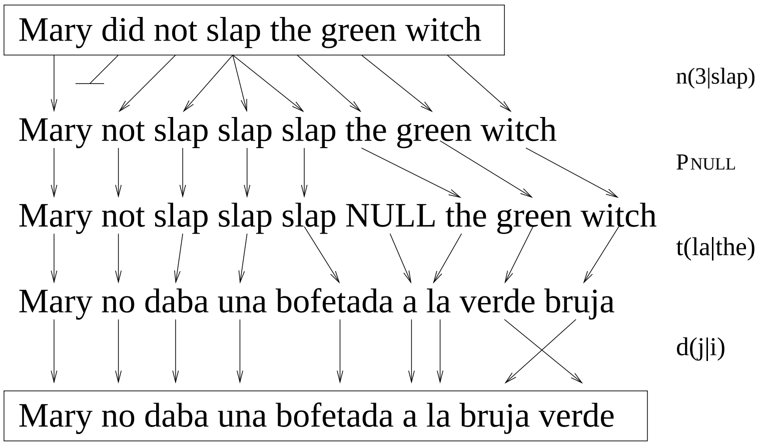


Figure 2.2: The translation process according to IBM Model 4 (illustration provided by Kevin Knight)

In the noisy channel framework, the probability $p(e|f)$ of the English translation e given an foreign input sentence f is reformulated using Bayes rule to $p(f|e)p(e)/p(f)$. This transformation allows for the use of a language model $p(e)$, which can be trained independently. This also means that the mathematical translation direction is reversed from $p(e|f)$ to $p(f|e)$. The factor $1/p(f)$ can be neglected in the search for the best translation e, since it is constant for a given input f.

The translation process is further decomposed into smaller steps that are modeled with probability distributions that are conditioned on single words. To illustrate this, consider the example in Figure 2.2.

The probability of the Spanish sentence given the English sentence is the product of a number of probabilities that model (1) word duplication, (2) word insertion, (3) word translation, and (4) word reordering. Each of the arrows in the example stands for a probability that is factored in. The resulting product is the sentence translation probability. This decomposition is mathematically motivated by marginalizing the joint probability distribution and a number of independence assumptions.

Strong independence assumptions limit the conditioning to only the directly affected words, hence enabling sufficient statistical basis for the estimation of the probability distribution from the data.

The different models proposed by the IBM group differ only in the conditioning of the probability distributions. For instance Model 4 uses relative movement (with respect to the previous word), while Models 1–3 use absolute word reordering. Later work replaces these probability distributions with maximum entropy classifiers that allow to take local context into account.

The decoding problem of finding an English output sentence for a given input foreign sentence is NP-complete for the IBM Models [Knight, 1999]. Thus, it requires search heuristics such as beam search [Och, 1998; Al-Onaizan et al., 1999; Och et al., 2001], integer programming, or greedy hill-climbing [Germann et al., 2001].

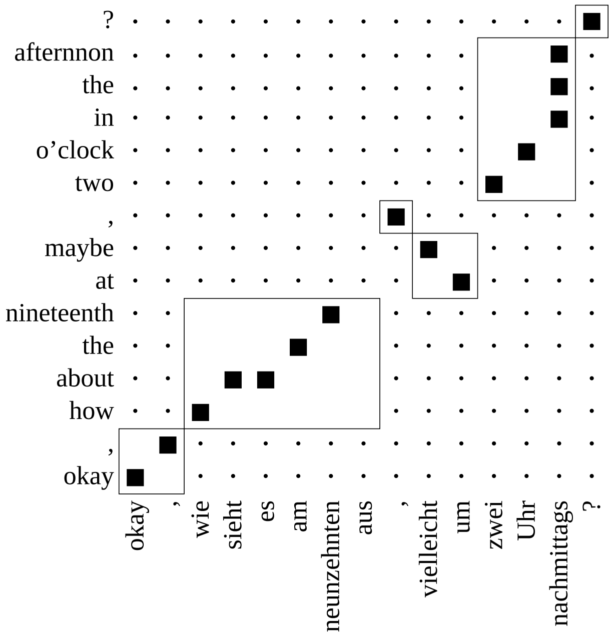


Figure 2.3: Alignment templates: Each framed box represents a template that defines reordering and translation of words of certain word classes (taken from Och [1998]).

2.2.2 Phrase-Based: Alignment Templates

One of the problems of the IBM models is that they do not allow for the translation of one single foreign word into multiple English words. This flaw is only somewhat overcome by the insertion of zero-fertility English words during decoding, e.g. the English word *did* in the previous example. However, these insertions are not sensitive to the context of the word they are applied to.

Och [1998] overcomes this problem by introducing alignment templates to replace the insertion and fertility probability distributions. This is illustrated by the example in Figure 2.3.

First the foreign (here, German) input sentence is segmented into small phrases. These phrases may or may not relate to syntactic units. For each phrase, an alignment template is chosen, which defines the class, alignment and placement of the English words to be generated. Finally, the English words are fleshed out.

An alignment template is defined as a local alignment matrix over word classes that are acquired by a bilingual clustering method [Och, 1999]. In other words, the alignment template selects the number and position of English words and limits the choices for English words to a specific word class for each position. Also, an alignment template may only match to a German phrase whose words have matching word classes.

In the special case of one word per word class, the alignment templates are practically aligned phrase pairs, with *phrase* simple meaning any sequence of words. Often, this simplified form of the alignment template model already yields the best performance.[1]

One can view this simplified form of the alignment template model as a phrase translation model similar to the one proposed by Marcu and Wong [2002]. The main difference between these models is the way, the phrases are learned: The alignment template model uses word alignments generated with the IBM Models, while Marcu and Wong's joint phrase model directly aligns phrases in the training corpus.

Phrase-based translation constitutes the state of the art in statistical machine translation, as performance in recent evaluations suggests.[2] Recently, alternative methods to learn phrase translation tables have been proposed [Tillmann, 2003; Venugopal et al., 2003] and compared [Koehn et al., 2003].

2.3 Syntax and Statistical Machine Translation

The basic statistical machine translation models we just described in the previous section make virtually no linguistic assumptions about the data.

However, there are many reasons to assume that some grasp of syntax should help to improve the quality of statistical machine translation. To note a few:

- Some transformations during translation (reordering, insertion and deletion of function words) can be best explained with syntactic concepts.
- Syntactic analysis of the input sentences provides additional knowledge that can be exploited.
- The use of syntax on the target side allows for the use of syntactic language models that help to ensure grammatical output of the system.

We will now describe two lines of research that exploit syntactic notions for statistical machine translation.

2.3.1 Syntax Tree Reordering

Syntax is represented in tree structures. So the translation process may be best explained not as a string transformation process, but as a tree transformation process. The intuition that word reordering during translation can be reduced to node reordering of syntactic trees is explicitly used in work by Wu [1997] on Inversion Transduction Grammars (ITG).

Wu does not use the noisy channel model, but a joint probability model that generates foreign and English sentences at once. The transduction grammar is automatically learned

[1] personal communication

[2] Phrase-based translation models showed the best performance in the large-corpus 2003 DARPA TIDES Chinese-English, Arabic-English evaluations.

from parallel corpora. It only permits binary nonterminal rules and terminal rules of the following form:

$$\begin{aligned} A &\rightarrow BC/BC \\ A &\rightarrow BC/CB \\ A &\rightarrow x/y \\ A &\rightarrow x/\epsilon \\ A &\rightarrow \epsilon/y \end{aligned}$$

Uppercase letters stand for non-terminals, lowercase for terminals. The grammar generates both English and Chinese, as indicated by the slash. As the formalism indicates, reordering is restricted to the non-terminal children of a node. Wu argues that this is sufficient for the Chinese/English corpus examined, but notes that this may not hold for free-word order languages [Wu, 1997, page 385].

Tree transfer models were also proposed by other researchers [Alshawi et al., 2000; Eisner, 2003; Gildea, 2003; Melamed, 2003].

2.3.2 Translation to Syntax Trees

The intuition that real syntax trees are beneficial for the task of machine translation and the recent availability of high accuracy statistical syntactic parsers is the basis of work by Yamada [2002]. He moves half-way to a translation model between real syntax trees by employing a parser on the English side of the corpus. On the foreign side, however, words are treated as tokens without any markup, as in previous work.

For training, the English side of the corpus is parsed with a high accuracy parser provided by Collins [1997]. From this data, a translation model is built that resembles a monolingual probabilistic parser, with additional node-level reordering rules and word-level translation rules. The decoding problem of translating a foreign input word sequence into an English parse tree becomes in effect a parsing problem, with additional reordering and word translation.

Besides word-translation on the leaf level, Yamada introduces phrasal translation of leaf nodes with the same parent. This enables, for instance, translation of base noun phrases directly without decomposing this step to word-level translation. However, such phrase translation is limited to constituents in the parse tree.

Since tree transfer is more complex than string transfer, it is also computationally more expensive and has not be shown to be practical for large scale domains such as news paper text. Still, this is a very active field of research at this time.

2.4 Defining Subtasks

There is some previous work that can portrayed as focusing on subtasks of the machine translation problem – as we do here with noun phrases.

2.4.1 Lexical Translation

One obvious subtask is lexical translation. Canonical statistical machine translation tables use one translation table with entries of the form $p(\text{foreign-word}|\text{english-word})$.

Disambiguation between the different lexical choices is based on the weighting in the lexical probability distribution and the language model. The language model provides some local context for disambiguation. Some researchers investigated, whether a wider context may be helpful. Research in word sense disambiguation, which is related to the problem of lexical choice, suggests that a window of up to 50 words surrounding an ambiguous word choice can provide useful clues.

Both Berger et al. [1996] and Varea et al. [2001] use features over words within a small window in the source and target language in a maximum entropy framework. Since word choice is integrated into the search of a decoding algorithm, it is computationally difficult to exploit larger window sizes. Varea et al. [2001] report no significant performance gains over flat lexical translation probability distributions.

In previous work, we addressed the question of how lexical translation tables can be learned in the absence of a parallel corpus. Using the EM algorithm, the lexical probability distribution can be learned for a non-probabilistic translation dictionary with the help of comparable monolingual corpora [Koehn and Knight, 2000]. A translation lexicon may also be learned from monolingual corpora alone, using clues such as spelling similarity, context vectors, and semantic similarity [Koehn and Knight, 2002b]. Koehn and Knight [2001] compare different knowledge sources for lexical translation in terms of machine translation performance.

We do not incorporate these methods for lexical translation in our work. Generally, lexical choice is not as hard as a problem for statistical machine translation as for other approaches, due to the use of a language model which uses local context to disambiguate word choices.

2.4.2 Named Entity Translation

Al-Onaizan and Knight [2002] addresses the subtask of the translation of named entities: names of places, people, places, organization, as well as dates, quantities, and numbers. Named entity translation is a big problem, especially if names have to translated from different scripts, such as Chinese, Arabic or Cyrillic. Al-Onaizan translates names using transliteration models, non-probabilistic translation lexicons, and web frequencies.

While the noun phrases we address in our work include named entities, we do not treat the specially, since they were no significant cause of error: Both English and German use the same Latin alphabet, so names can be reproduced verbatim.

2.4.3 BaseNP Translation

Cao and Li [2002] address the translation of base noun phrases. Their work is restricted to noun-noun pairs that are translated into noun-noun pairs. Translation candidates are ranked using a non-probabilistic translation lexicon and large monolingual corpora (in fact, the web). The methodology is very similar to our earlier work [Koehn and Knight, 2000]: the EM algorithm is applied to learn lexical translation probabilities and context vector similarity is used for additional disambiguation.

Compared with our work, Cao and Li address a much more limited problem with methods that are suited to such a limited problem. We are dealing with more complex noun phrases (recall Figure 1.1 from page 2).

2.5 Conclusion

The trajectory of the statistical machine translation work just described reflects the intuition that, for machine translation, syntactic structure is important. Previous work is limited to the automatic acquisition of transfer mechanisms from parallel corpora that have no additional linguistic markup – with the exception of Yamada [2002], who uses full parses in the target language.

The lack of linguistic markup employed had foremost practical reasons: The necessary tools to provide this markup were not available at the time. Indeed, the direction toward more syntactic models of the translation process underlines the common intuition that linguistically more sophisticated models are key to better performance. We will attempt this in the proposed work.

Chapter 3

Framework

This chapter describes the design of the system and its underlying mathematical models that are the foundations of this work.

3.1 Overview

Let us start with a brief overview of the system design.

3.1.1 Dedicated NP/PP Translation

Figure 3.1 visualizes the design of our system: When translating a foreign input sentence, we detect its NP/PPs and translate them with an NP/PP translation subsystem. The best translation (or a set of best translations) is then passed on to the full sentence translation system which in turn translates the remaining parts of the sentence while integrating the chosen NP/PP translations.

In other words, the process of machine translation with a dedicated NP/PP translation system takes place in three steps:

Detection of NP/PPs Each input sentence is scanned for NP/PPs. For this, the input has to be POS tagged and syntactically parsed. This detection of NP/PPs follows straight forward from the definition for NP/PP (see Section 3.2.1.1 for more detail on detecting NP/PPs).

NP/PP Subsystem Each NP/PP is translated separately. This chapter describes the framework of the NP/PP subsystem in more detail.

Integration The translations provided by the NP/PP subsystem have to be integrated into a general translation system that translates the rest of the sentence. Integration is described in Chapter 5.

3.1.2 Overview of NP/PP Subsystem

Our NP/PP translation subsystem (see also Figure 3.2) follows a reranking approach based on a base model. The base model proposes a list of candidate translations that are reranked with additional features.

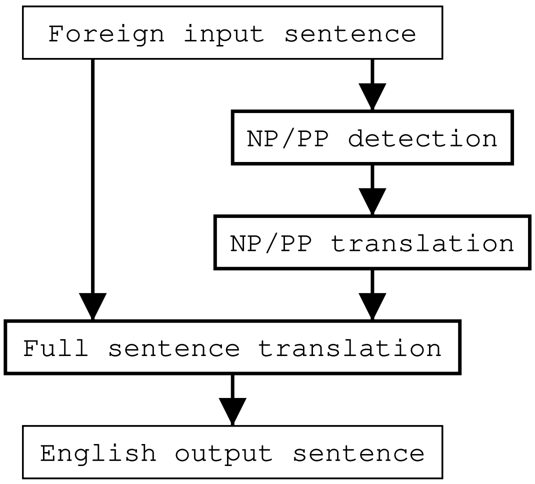


Figure 3.1: System Design: NP/PPs of the input sentence are detected and translated by a separate NP/PP translation subsystem. The full sentence translation system integrates the NP/PP translations with the translations of the rest of the sentence.

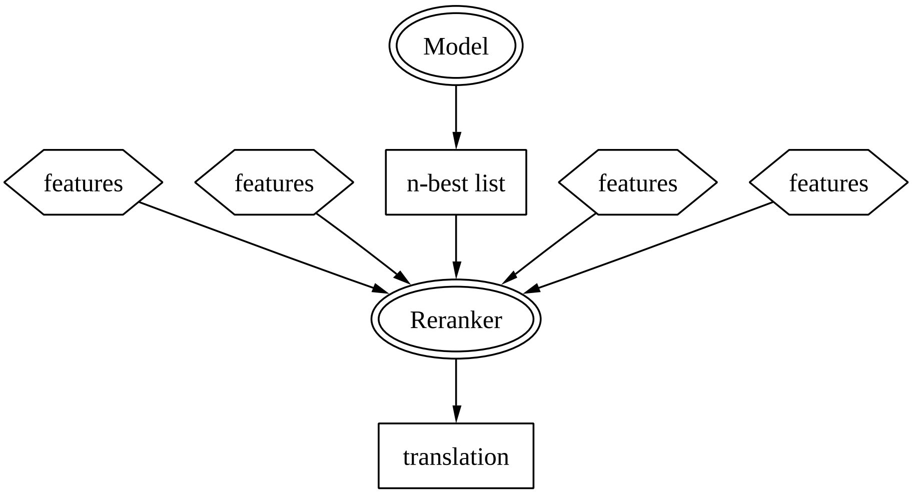


Figure 3.2: Design of the noun phrase translation subsystem: the base model generates an n-best list that is rescored using additional features

We train a base statistical translation model on a NP/PP parallel corpus. The acquisition of such a corpus is described in Section 3.2. The base model and its decoding algorithm is described in Section 3.3.

We use the trained base model to generate an n-best list of possible translations. A discussion of the construction and size of these n-best lists is given in Section 3.4.

We then rerank this n-best list with the help of additional features. This reranking is a standard supervised machine learning problem, for which many methods exist. We chose maximum likelihood for its successful history in natural language tasks and its ability to work well with a heterogeneous set of features (binary, integer, real numbered). The maximum entropy machine learning framework for these features is presented in Section 3.5. Chapter 4 discusses in depth the properties of NP/PP translation that are exploited to improve translation performance.

3.2 Acquisition of an NP/PP Corpus

To train a statistical machine translation model, we need a training corpus of NP/PPs paired with their translation. We create this corpus by extracting NP/PPs from a sentence-aligned parallel corpus (a text along with its translation).

In this section, we describe the acquisition of such an NP/PP corpus. We also provide experimental results and other specific findings for extracting a large German-English NP/PP corpus from the Europarl corpus.

3.2.1 Detecting and Aligning NP/PPs

To align NP/PPs in a parallel corpus, we need two elements: we need to know (1) which words group together to form NP/PPs, and (2) which words on the foreign sides align to which words on the English side.

3.2.1.1 Detecting NP/PPs

To be able to detect NP/PPs in a corpus, we first annotate it with syntactic parse trees. We obtain these syntactic parse trees by using state-of-the-art statistical syntactic parsers. For English we use the well-known statistical parser by Collins [1997]; for German we use the statistical parser LoPar [Schmidt and Schulte im Walde, 2000].

Our definition of NP/PPs translates quite easily into an algorithm to mark up NP/PPs given a syntactic parse tree:

- For each node in the tree we determine if it contains at least a noun. These are potential NP/PPs.
- We then determine for each potential NP/PP if it contains a verb. Those that do are eliminated as potential NP/PPs.
- Finally, since we are looking for the maximal NP/PPs, we eliminate each node that has a parent node that is also a potential NP/PP.

3.2.1.2 Aligning NP/PPs

We establish the alignment of NP/PPs by relying on a given word alignment. There are many known methods for establishing word alignments for a parallel corpus.

Note that instead of following the strategy of aligning detected NP/PPs with the use of a word alignment, one may also devise a method that directly aligns NP/PPs between sentences. We did not pursue this direction.

In our work, we follow a strategy proposed by Och and Ney [2000] to obtain a word alignment: First, we train the IBM Model 4 machine translation models on the parallel corpus, using the Giza++ toolkit. We do this bidirectionally, foreign-English and English-foreign. Each training run produces a word alignment. We reconcile the two word alignments with heuristics described in detail in Section B.4.

At this point, we have a word alignment of the parallel corpus, and we also detected the NP/PPs on each side of the corpus. This enables us to extract NP/PPs from the corpus as follows: a foreign NP/PP is aligned to an English NP/PP if all the words in the foreign NP/PP are only aligned to words in the English NP/PP, and vice versa.

Every foreign NP/PP that does not align to one single English NP/PP is discarded. The same is done for improperly aligned English NP/PP. Thus, only correctly aligned NP/PP pairs are stored in our NP/PP corpus.

3.2.2 Data Cleaning

We cannot expect that all foreign NP/PPs align neatly to English NP/PPs. Recall that in a parallel corpus, some of the NP/PPs are not translated into NP/PPs in the foreign language (see Section 1.6 where we observed that 75% of NP/PPs are aligned). In addition, the word alignments and syntactic parses may be faulty. As a consequence, initially only 43.4% of all NP/PPs could be aligned. We raise this number to 67.2% with a number of data cleaning steps.

Some of the data cleaning steps are specific the German-English language pair, some are more general. We perform the following steps:

- NP/PPs that are partially aligned are broken up.
- Systematic parse errors are fixed.
- Certain word types that are inconsistently tagged as nouns in the two languages are harmonized (e.g. the German *wo* and the English *today*).
- Since adverb + NP/PP constructions (e.g. *specifically this issue*) are inconsistently parsed, we always strip the adverb from these constructions.
- Certain German construction involving verbal adjectives are broken up, since they usually translate as relative clauses into English (e.g. *der von mir gegessene Kuchen* $\sim$ *the by me eaten cake* $\rightarrow$ *the cake eaten by me*).
- Alignment points involving punctuation are stripped from the word alignment. Punctuation is also stripped from the edges of NP/PPs.

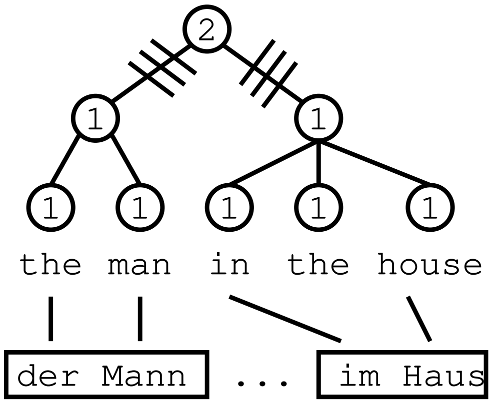


Figure 3.3: Breaking up partially aligned NP/PP: Nodes are annotated with how many NP/PPs they are aligned with. If more than one, they are removed, leaving (in this example) two NP/PPs that are uniquely aligned.

3.2.2.1 Breaking up Partially Aligned NP/PPs

Often, a German NP/PP is not aligned to a single English NP/PP, but to multiple NP/PPs. If we break up the German NP/PP so each of its parts aligns to only one English NP/PP, we preserve some valuable training data for our NP/PP corpus.

To accomplish this, we iteratively break up German and English NP/PPs that do not properly align. Candidates for break-up are NP/PPs that align to more than one NP/PP on the other side. Possible break-up points are nodes in the syntactic tree. This means that the parts are constituents in the syntactic parse tree.

The process is visualized in Figure 3.3. For each NP/PP that aligns to more than one NP/PP on the other side, we perform the following steps:

- We determine for each node in the syntactic tree of the NP/PP to how many different NP/PPs on the other side it aligns.

- We remove all nodes that align to more than one NP/PP. This leaves us with a number of smaller trees, each of which aligns to at most one NP/PP.

- Each partial tree that contains at least one noun is then classified as a NP/PP and matched with its counterpart on the other side.

A break-up of a German NP/PP might force the break-up of an English NP/PP and vice versa. So, we have to iteratively perform break-ups of German and English NP/PP until no more changes are warranted.

Note that after this processing step, no NP/PP is aligned anymore to more than one NP/PP. It is still possible, that (i) a NP/PP is not aligned to any NP/PP, and that (ii) a NP/PP is aligned to one single NP/PP, but also to other words that are not part of a NP/PP. Those erroneous NP/PPs are still discarded from the corpus.

3.2.2.2 Systematic Parse Errors

As a general assessment, the quality of the syntactic parsers used does impact the performance of our approach significantly. For the correct detection of NP/PPs, hard problems in parsing such as prepositional phrase attachment have to be addressed reliably. Unfortunately this is not always the case in the tools we used for our experiments.

Since the statistical parsers are trained on different material (mostly news sources), some idiosyncrasies of our corpora are not resolved correctly. We put some effort into post-processing the output of the statistical parsers. One example is the grouping of *Herr Kollege Smith* (English: *Mr. colleague Smith*), which the original German parser mistakenly breaks into multiple noun phrases attached at the clause level.

3.2.2.3 Harmonizing Definition of Noun

What constitutes a noun depends on the underlying linguistic theory. For instance, one might debate if *today* in its typical use is an adverb or a noun, or both. In the Penn treebank [Marcus et al., 1994], it is consistently tagged as a noun (NN). However, the German equivalent, *heute*, is tagged as a adverb (ADV) in the TIGER treebank [Brants et al., 2002].

To establish consistency across the two languages, we changed the designated part-of-speech for some words. For English, we consider the following words not as nouns: *today, everything, anything, something, nothing, tomorrow, someone, anyone, everyone, nobody.* For German, we consider the following words not as nouns: *man, wonach, womit, worum, wann, wo, warum, wer, wodurch, wobei, dar.*

3.2.2.4 Adverb + NP/PP Constructions

Consider the following two sentences:

- *He addressed specifically this issue.*
- *Es geht ihm genau um diesen Punkt.*

Syntactic theories vary whether the adverbs *specifically* and *genau* attach to the NP/PP (as indicated by the underlining above) or directly to the clause level. In a way, it is a matter of emphasis. These variations in German and English parses create a number of misaligned NP/PPs.

We resolve this problem by consistently attaching these adverbs to the clause level.

3.2.2.5 German Verbal Adjective Constructions

Another problematic issue is the role of German verbal adjectives, as illustrated by the following sentences:

- *Die am Donnerstag unterbrochene Sitzung ist wiederaufgenommen.*
  English gloss: *The on Thursday interrupted meeting is resumed.*
  English translation: *The meeting interrupted on Thursday is resumed.*

- *Der von mir gegessene Kuchen ist lecker.*
  English gloss: *The by me eaten cake is delicious.*
  English translation: *The cake eaten by me is delicious.*

The verbal adjectives *unterbrochene* (English: *interrupted*) and *gegessene* (English: *eaten*) may still take adjuncts and arguments as verbs in German, which is generally not possible in English. In the given examples, we underlined the constituents that are headed by an verbal adjective. In the literal English translation of the examples, the sentence structure is changed and the verbal adjectives are translated as verbs that introduce a relative clause following the noun. It is almost never possible to translate a verbal adjective that takes arguments or adjuncts as an adjective into English.

For this reason, we classify the verbal adjectives as a type of verb if they take adjuncts or arguments. This forces the NP/PP detection algorithm to disallow such constructions within NP/PPs.

3.2.2.6 Punctuation

The treebanks we used – and therefore the statistical syntactic parsers trained from it – are somewhat careless with regard to punctuation. For instance, the sentence ending period is often attached to the last noun phrase at the end of the sentence. Also, commas may show up at the beginning of a noun phrase constituent, when their correct syntactic position would be higher up in the syntactic tree.

Since we do not want to learn NP/PP translation pairs with inconsistently attached punctuation, we remove all punctuation from the beginning and end of NP/PPs.

Punctuation is also often falsely aligned to actual words. We remove all such alignments to eliminate this noise.

3.2.2.7 Evaluation of Data Cleaning

To evaluate the data cleaning steps, we carried out experiments and measured quantity and quality of the obtained NP/PP corpus.

Table 3.1 lists the number of aligned NP/PP pairs after different data cleaning stages, along with the different types of erroneously aligned NP/PPs: a German NP/PP may not be aligned to any English NP/PP (**Unaligned**), it may be aligned to multiple English NP/PP (**Multiple**), and it might be aligned to one or many English NP/PPs and in addition to other words in the English sentence that are not part of any NP/PP (**With outside**).

Note that not all unaligned NP/PPs represent errors of the acquisition process. If, say, the NP/PP *der Präsident* is aligned to the pronoun *he*, we want to exclude this example from the NP/PP corpus, since it is not a useful translation pair for our purposes. In this case, the unaligned NP/PP is the result of successful filtering.

The ratio of correctly aligned NP/PPs rises with each additional data cleaning step. The absolute numbers change due to the increasingly refined working definition of NP/PP. For instance, after reclassifying a number of nouns as non-nouns (harmonizing nouns, see Section 3.2.2.3), the total number of NP/PPs drops.

Cleaning Steps	Aligned		Unaligned		Multiple		With outside	
no cleaning	444,166	43.4%	154,449	15.1%	219,641	21.4%	205,518	20.1%
fix parse errors	459,493	46.0%	150,042	15.0%	187,372	18.7%	201,457	20.2%
+ break up partial	721,797	64.9%	171,748	15.4%	-	-	219,439	19.7%
+ harmonize nouns	719,407	66.1%	149,629	13.7%	-	-	219,776	20.2%
+ strip adverbs	729,993	66.9%	151,314	13.9%	-	-	209,490	19.2%
+ verbal adjectives	736,125	67.1%	155,793	14.2%	-	-	204,879	18.7%
+ strip punctuation	737,388	67.2%	155,782	14.2%	-	-	203,625	18.6%

Table 3.1: Evaluation of the data cleaning steps: more cleanly aligned NP/PP (aligned) pairs are collected. The number of erroneously aligned NP/PP (unaligned, multiple, with outside) that cannot be included in the NP/PP corpus is reduced.

Some of the data cleaning steps are specific to this language pair (e.g., dealing with verbal adjectives), while others are general (e.g., breaking up partial NP/PPs). Most steps can be performed similarly for other language pairs (e.g., harmonizing nouns).

Overall we raised the ratio of correctly aligned German NP/PPs from 43.4% to 67.2%. The size of the acquired increased from 444,166 NP/PP pairs to 737,388 NP/PP pairs.

3.2.3 Unaligned NP/PP

Even after data cleaning, a large portion of the German NP/PPs (32.8%) do not align cleanly to English NP/PPs. They align to words that are not part of any NP/PP, or to an English NP/PP with additional words outside.

Some of these unaligned NP/PPs are noise that we successfully filtered out (e.g., *der Präsident* = *he*). Some are German NP/PPs that systematically do not align to English NP/PPs (recall the examples in Section 1.6). These illustrate cases where verb+NP/PP constructions are best translated as verbs (*make an observation* = *observe*), and NP/PPs that translate as adverbs (*in the main thing* = *mainly*).

During the acquisition of the NP/PP corpus, these special cases can be detected and recorded. This enables the construction of a classifier that decides which NP/PP are to be translated as NP/PP, and which are to be handled specially. For examples such as the ones given in Section 1.6, a simple memorization approach may be sufficient. We did not pursue work in this area, since it is not a very significant problem in terms of overall performance. Recall that we expect to translate roughly 98% of the NP/PPs as NP/PPs, so we did not focus on the remaining 2%.

3.3 Base Model

The base model is used to generate n-best lists of candidate translations that are reranked at a later stage with the aid of additional features (see Chapter 4). Our base model is a statistical phrase-based translation model, which is similar to Och et al. [1999]'s alignment template model. Instead of using word classes and alignment templates, however, we use directly a phrase translation table. This allows us to build a more compact, more transparent, and faster decoder, without loss in performance.

This section describes this model in detail and compares it against other phrase-based approaches. Since the model is designed for general statistical machine translation, the experiments in this section are carried out on full sentence translation. The core results hold also for NP/PP translation, as separate experiments show. We also use the model for full sentence translation when integrating output of the NP/PP subsystem (see Chapter 5).

Based on the findings presented in this section and results of a recent international machine translation competition[1], we can state that, currently, our phrase-based translation model follows the best-known approach to statistical machine translation.

3.3.1 Phrase-Based Translation

The original statistical machine translation approach Brown et al. [1990] maps individual words to words. Various researchers have improved the quality of statistical machine translation systems with the use of phrase translation. Och et al. [1999]'s alignment template model can be re-framed as a phrase translation system; Yamada and Knight [2001] use phrase translation in a syntax-based translation system; Marcu and Wong [2002] introduced a joint-probability model for phrase translation; and the recent CMU [Venugopal et al., 2003] and IBM [Tillmann, 2003] statistical machine translation systems use phrase translation.

Phrase translation clearly helps, as we will also show with the experiments in this thesis. But what is the best method to extract phrase translation pairs? In order to investigate this question, we created a uniform evaluation framework that enables the comparison of different ways to build a phrase translation table.

Our experiments show that high levels of performance can be achieved with fairly simple means. In fact, for most of the steps necessary to build a phrase-based system, tools and resources are freely available for researchers in the field. More sophisticated approaches that make use of syntax do not yet lead to better performance. In fact, imposing syntactic restrictions on phrases, as used in recently proposed syntax-based translation models [Yamada and Knight, 2001], proves to be harmful. Our experiments also show that small phrases of up to three words are sufficient for obtaining high levels of accuracy.

Performance differs widely depending on the methods used to build the phrase translation table. We found extraction heuristics based on word alignments to be better than a more principled phrase-based alignment method. However, what constitutes the best heuristic differs from language pair to language pair and varies with the size of the training corpus.

[1] DARPA TIDES Machine Translation Evaluation 2003 on Chinese-English and Arabic-English

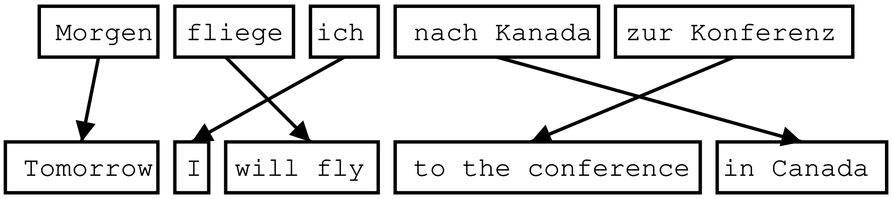


Figure 3.4: Phrase-based machine translation: input is segmented in phrases, each is translated and may be reordered.

3.3.2 Model

Figure 3.4 illustrates the process of phrase-based translation. The input is segmented into a number of sequences of consecutive words (so-called "phrases"). Each phrase is translated into an English phrase, and English phrases in the output may be reordered.

The phrase translation model is based on the noisy channel model. We use Bayes rule to reformulate the translation probability for translating a foreign sentence $\mathbf{f}$ into English $\mathbf{e}$ as

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p(\mathbf{e})$$

This allows for a language model $p(\mathbf{e})$ and a separate translation model $p(\mathbf{f}|\mathbf{e})$.

During decoding, the foreign input sentence $\mathbf{f}$ is segmented into a sequence of I phrases $\bar{f}_1^I$. We assume a uniform probability distribution over all possible segmentations.

Each foreign phrase $\bar{f}_i$ in $\bar{f}_1^I$ is translated into an English phrase $\bar{e}_i$. The English phrases may be reordered. Phrase translation is modeled by a probability distribution $\phi(\bar{f}_i|\bar{e}_i)$. Recall that due to the Bayes rule, the translation direction is inverted from a modeling standpoint.

Reordering of the English output phrases is modeled by a relative distortion probability distribution $d(a_i - b_{i-1})$, where a_i denotes the start position of the foreign phrase that was translated into the ith English phrase, and b_{i-1} denotes the end position of the foreign phrase translated into the $(i-1)$th English phrase.

In all our experiments, the distortion probability distribution $d(\cdot)$ is trained using a joint probability model (see Section 3.3.4.3). Alternatively, we could also use a simpler distortion model $d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|}$ with an appropriate value for the parameter α.

In order to calibrate the output length, we introduce a factor ω (called word cost, see also Appendix B.3) for each generated English word in addition to the trigram language model p_{LM}. This is a simple means to optimize performance. Usually, this factor is larger than 1, biasing toward longer output.

In summary, the best English output sentence $\mathbf{e}_{\text{best}}$ given a foreign input sentence $\mathbf{f}$ according to our model is

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f})$$

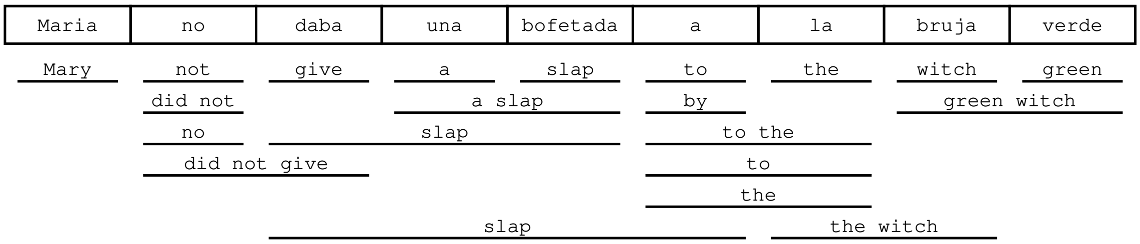


Figure 3.5: Some translation options for the Spanish input sentence *Maria no daba una bofetada a la bruja verde*

$$= \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p_{\text{LM}}(\mathbf{e}) \omega^{\text{length}(\mathbf{e})}$$

where $p(f|e)$ is decomposed into

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

For all our experiments we use the same training data, trigram language model [Seymore and Rosenfeld, 1997], and the same decoder described in the next section.

3.3.3 Decoder

Phrase-based translation in similar form has been previously proposed by Marcu and Wong [2002]. However, only a greedy decoding algorithm is known and available for this model. Since we need a decoding algorithm that generates n-best translation lists, we propose a novel decoder for phrase-based machine translation. The decoder implements a beam search and is roughly similar to work by Tillmann [2001] and Och [2002].

3.3.3.1 Translation Options

Given an input string of words, a number of phrase translations could be applied. We call each such applicable phrase translation a *translation option*. This is illustrated in Figure 3.5. Here, a number of phrase translations for the Spanish input sentence *Maria no daba uma bofetada a la bruja verde* are given.

These translation options are collected before any decoding takes place. This allows a quicker lookup than consulting the whole phrase translation table during decoding. The translation options are stored with the information

- first foreign word covered
- last foreign word covered
- English phrase translation
- phrase translation probability

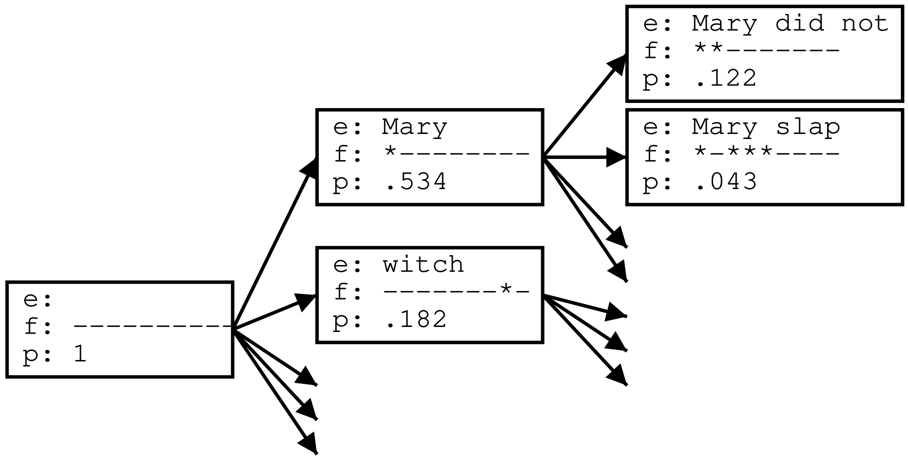


Figure 3.6: State expansion in the beam decoder: in each expansion English words are generated, additional foreign words are covered (marked by *), and the probability cost so far is adjusted. In this example the input sentence is *Maria no daba una bofetada a la bruja verde.*

Note that only the translation options that can be applied to a given input text are necessary for decoding. Since the entire phrase translation table may be to big to fit into memory, we can restrict ourselves to these translation options to overcome such computational concerns. We may even generate a phrase translation table on demand that only includes valid translation options for a given input text. This way, a full phrase translation table (that may be computationally too expensive to produce) may never have to be built.

3.3.3.2 Core Algorithm

The phrase-based decoder we developed employs a beam search algorithm, similar to the one used by [Jelinek, 1998, Chapter 5] for speech recognition. The English output sentence is generated left to right in form of hypotheses.

This process illustrated in Figure 3.6. Starting from the initial hypothesis, the first expansion is the foreign word *Maria*, which is translated as *Mary*. The foreign word is marked as translated (marked by an asterisk). We may also expand the initial hypothesis by translating the foreign word *bruja* as *witch*.

We can generate new hypotheses from these expanded hypotheses. Given the first expanded hypothesis we generate a new hypothesis by translating *no* with *did not*. Now the first two foreign words *Maria* and *no* are marked as being covered. Following the back pointers of the hypotheses we can read of the (partial) translations of the sentence.

Let us now describe the beam search more formally. We begin the search in an initial state where no foreign input words are translated and no English output words have been generated. New states are created by extending the English output with a phrasal translation of that covers some of the foreign input words not yet translated.

The current cost of the new state is the cost of the original state multiplied with the translation, distortion and language model costs of the added phrasal translation. Note that we use the informal concept *cost* analogous to probability: A high cost is a low probability.

Each search state (hypothesis) is represented by

- a back link to the best previous state (needed for find the best translation of the sentence by back-tracking through the search states)
- the foreign words covered so far
- the last two English words generated (needed for computing future language model costs)
- the end of the last foreign phrase covered (needed for computing future distortion costs)
- the last added English phrase (needed for reading the translation from a path of hypotheses)
- the cost so far
- an estimate of the future cost (is precomputed and stored for efficiency reasons, as detailed in Section 3.3.3.5)

Final states in the search are hypotheses that cover all foreign words. Among these the hypothesis with the lowest cost (highest probability) is selected as best translation.

The algorithm described so far can be used for exhaustively searching through all possible translations. In the next sections we will describe how to optimize the search by discarding hypotheses that cannot be part of the path to the best translation. We then introduce the concept of comparable states that allow us to define a beam of good hypotheses and prune out hypotheses that fall out of this beam. In a later section (Section 3.4), we will describe how to generate an (approximate) n-best list.

3.3.3.3 Recombining Hypotheses

Recombining hypothesis is a risk-free way to reduce the search space. Two hypotheses can be recombined if they agree in

- the foreign words covered so far
- the last two English words generated
- the end of the last foreign phrase covered

If there are two paths that lead to two hypotheses that agree in these properties, we keep only the cheaper hypothesis, e.g., the one with the least cost so far. The other hypothesis cannot be part of the path to the best translation, and we can safely discard it.

Note that the inferior hypothesis can be part of the path to the second best translation. This is important for generating n-best lists. We return to this point in Section 3.4.1.

3.3.3.4 Beam Search

While the recombination of hypotheses as described above reduces the size of the search space, this is not enough for all but the shortest sentences. Let us estimate how many hypotheses (or, states) are generated during an exhaustive search. Considering the possible values for the properties of unique hypotheses, we can estimate an upper bound for the number of states by

$$N \simeq 2^{n_f} |V_e|^2 n_f \tag{3.1}$$

where n_f is the number of foreign words, and $|V_e|$ the size of the English vocabulary. In practice, the number of possible English words for the last two words generated is much smaller than $|V_e|^2$. The main concern is the exponential explosion from the 2^{n_f} possible configurations of foreign words covered by a hypothesis. Note this causes the problem of machine translation to become NP-complete [Knight, 1999] and thus dramatically harder than, for instance, speech recognition.

In our beam search we compare the hypotheses that cover the same *number* of foreign words and prune out the inferior hypotheses. We could base the judgment of what inferior hypotheses are on the cost of each hypothesis so far. However, this is generally a very bad criterion, since it biases the search to first translating the easy part of the sentence. For instance, if there is a three word foreign phrase that easily translates into a common English phrase, this may carry much less cost than translating three words separately into uncommon English words. The search will prefer to start the sentence with the easy part and discount alternatives too early.

So, our measure for pruning out hypotheses in our beam search does not only include the cost so far, but also an estimate of the future cost. This future cost estimation should favor hypotheses that already covered difficult parts of the sentence and have only easy parts left, and discount hypotheses that covered the easy parts first. We describe the details of our future cost estimation in the next section.

Given the cost so far and the future cost estimation, we can prune out hypotheses that fall outside the beam. The beam size can be defined by threshold and histogram pruning. A relative threshold cuts out a hypothesis with a probability less than a factor α of the best hypotheses (e.g., $\alpha = 0.001$). Histogram pruning keeps a certain number n of hypotheses (e.g., $n = 1000$).

Note that this type of pruning is not risk-free (opposed to the recombination, which we described earlier in Section 3.3.3.3). If the future cost estimates are too far off, we may prune out hypotheses on the path to the best scoring translation. In a particular version of beam search, A* search, the future cost estimate is required to be *admissible*, which means that it never overestimates the future cost. Using best-first search and an admissible heuristic allows pruning that is risk-free. In practice, however, this type of

```
initialize hypothesisStack[0 .. nf];
create initial hypothesis hyp_init;
add to stack hypothesisStack[0];
for i=0 to nf-1:
  for each hyp in hypothesisStack[i]:
    for each new_hyp that can be derived from hyp:
      nf[new_hyp] = number of foreign words covered by new_hyp;
      add new_hyp to hypothesisStack[nf[new_hyp]];
      prune hypothesisStack[nf[new_hyp]];
find best hypothesis best_hyp in hypothesisStack[nf];
output best path that leads to best_hyp;
```

Figure 3.7: Pseudo code for the beam search algorithm

pruning does not sufficiently reduce the search space. See more on search in any good Artificial Intelligence text book, such as the one by Russel and Norvig [1995].

Figure 3.7 describes the algorithm we used for our beam search. For each number of foreign words covered, a hypothesis stack in created. The initial hypothesis is placed in the stack for hypotheses with no foreign words covered. Starting with this hypothesis, new hypotheses are generated by committing to phrasal translations that covered previously unused foreign words. Each derived hypothesis is placed in a stack based on the number of foreign words it covers.

We proceed through these hypothesis stacks, going through each hypothesis in the stack, deriving new hypotheses for this hypothesis and placing them into the appropriate stack (see Figure 3.8 for an illustration). After a new hypothesis is placed into a stack, the stack may have to be pruned by threshold or histogram pruning, if it has become too large. In the end, the best hypothesis of the ones that cover all foreign words is the final state of the best translation. We can read off the English words of the translation by following the back links in each hypothesis.

3.3.3.5 Future Cost Estimation

Recall that for excluding hypotheses from the beam we do not only have to consider the cost so far, but also an estimate of the future cost. While it is possible to calculate the cheapest possible future cost for each hypothesis, this is computationally so expensive that it would defeat the purpose of the beam search.

The future cost is tied to the foreign words that are not yet translated. In the framework of the phrase-based model, not only may single words be translated individually, but also consecutive sequences of words as a phrase.

Each such translation operation carries a translation cost, a language model cost, and a distortion cost. For our future cost estimate we consider only translation and language model costs. The language model cost is usually calculated by a trigram language model. However, we do not know the preceding English words for a translation operation. Therefore, we approximate this cost by computing the language model score for the generated

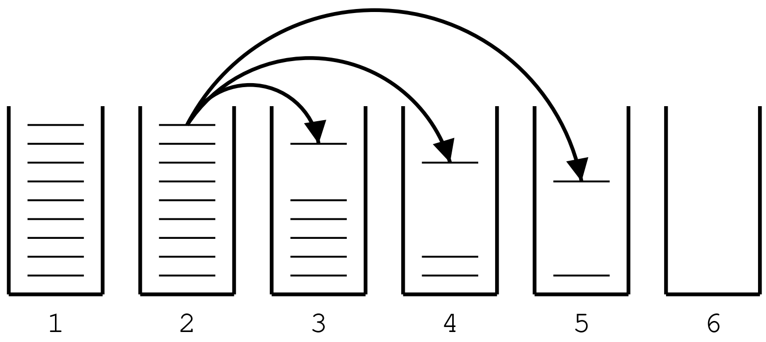


Figure 3.8: Hypothesis expansion: Hypotheses are placed in stacks according to the number of foreign words translated so far. If a hypothesis is expanded into new hypotheses, these are placed in new stacks.

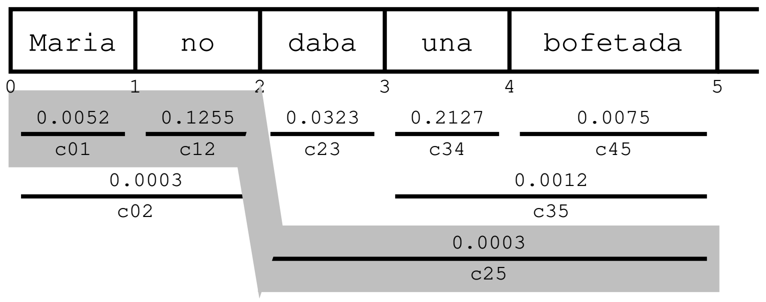


Figure 3.9: Finding the best future cost path through translation options. The cheapest cost is $c_{01}c_{12}c_{25} = 0.0052 \times 0.1255 \times 0.0003 = 1.9578 \times 10^{-7}$, hence it is the estimate of the cost of translating the five words *Maria no daba una bofetada*.

English words alone. That means, if only one English word is generated, we take its unigram probability. If two words are generated, we take the unigram probability of the first word and the bigram probability of the second word, and so on.

For a sequence of foreign words multiple overlapping translation options exist. We just described how we calculate the cost for each translation option. The cheapest way to translate the sequence of foreign words includes the cheapest translation options. We approximate the cost for a path through translation options by the product of the cost for each option.

To illustrate this concept, refer to Figure 3.9. The translation options cover different consecutive foreign words and carry an estimated cost c_{ij}. The cost of the shaded path through the sequence of translation options is $c_{01}c_{12}c_{25} = 1.9578 \times 10^{-7}$.

The cheapest path for a sequence of foreign words can be quickly computed with dynamic programming. Also note that if the foreign words not covered so far are two (or more) disconnected sequences of foreign words, the combined cost is simply the product of the costs for each contiguous sequence. Since there are only $n(n+1)/2$ contiguous sequences for n words, the future cost estimates for these sequences can be easily precomputed and cached for each input sentence. Looking up the future costs for a hypothesis can then be done very quickly by table lookup. This has considerable speed advantages over computing future cost on the fly.

This concludes our description of the decoder for now. We describe additional features of the decoder later in this thesis: the generation of n-best lists (Section 3.4) and a XML markup scheme that allows us to integrate the NP/PP subsystem into a general machine translation system (Section 5.2).

3.3.4 Methods for Learning Phrase Translation

When describing the phrase-based translation model so far, we did not discuss how to obtain the model parameters, especially the phrase probability translation table that maps foreign phrases to English phrases.

We carried out experiments to compare the performance of three different methods to build phrase translation probability tables. We also investigate a number of variations. We report most experimental results on a German-English full sentence translation task, since we had sufficient resources available for this language pair. We confirm the major points in experiments on additional language pairs.

As the first method, we learn phrase alignments from a corpus that has been word-aligned by a training toolkit for a word-based translation model: the Giza++ toolkit [Och and Ney, 2000] for the IBM models [Brown et al., 1993]. The extraction heuristic is similar to the one used in the alignment template work by Och et al. [1999].

A number of researchers have proposed focusing on the translation of phrases that have a linguistic motivation [Yamada and Knight, 2001; Imamura, 2002]. They only consider word sequences as phrases if they are constituents, i.e., subtrees in a syntax tree (such as a noun phrase). To identify these, we use a word-aligned corpus annotated with parse trees generated by statistical syntactic parsers [Collins, 1997; Schmidt and Schulte im Walde, 2000].

The third method for comparison is the joint phrase model proposed by Marcu and Wong [2002]. This model learns directly a phrase-level alignment of the parallel corpus.

3.3.4.1 Phrases from Word-Based Alignments

The Giza++ toolkit was developed to train word-based translation models from parallel corpora. As a by-product, it generates word alignments for this data. We improve this alignment with a number of heuristics.

We collect all aligned phrase pairs that are consistent with the word alignment: The words in a legal phrase pair are only aligned to each other, and not to words outside [Och et al., 1999].

Given the collected phrase pairs, we estimate the phrase translation probability distribution by relative frequency:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f}, \bar{e})}$$

No smoothing is performed.

3.3.4.2 Syntactic Phrases

If we collect all phrase pairs that are consistent with word alignments, this includes many non-intuitive phrases. For instance, translations for phrases such as "house the" may be learned. Intuitively we would be inclined to believe that such phrases do not help: Restricting possible phrases to syntactically motivated phrases could filter out such non-intuitive pairs.

Another motivation to evaluate the performance of a phrase translation model that contains only syntactic phrases comes from recent efforts to built syntactic translation models [Yamada and Knight, 2001; Wu, 1997]. In these models, reordering of words is restricted to reordering of constituents in well-formed syntactic parse trees. When augmenting such models with phrase translations, typically only translation of phrases that span entire syntactic subtrees is possible. It is important to know if this is a helpful or harmful restriction.

Consistent with Imamura [2002], we define a syntactic phrase as a word sequence that is covered by a single subtree in a syntactic parse tree.

We collect syntactic phrase pairs as follows: we word-align a parallel corpus, as described in Section 3.3.4.1. We then parse both sides of the corpus with syntactic parsers [Collins, 1997; Schmidt and Schulte im Walde, 2000]. For all phrase pairs that are consistent with the word alignment, we additionally check if both phrases are subtrees in the parse trees. Only these phrases are included in the model.

Hence, the syntactically motivated phrase pairs learned are a subset of the phrase pairs learned without knowledge of syntax (Section 3.3.4.1).

As in Section 3.3.4.1, the phrase translation probability distribution is estimated by relative frequency.

3.3.4.3 Phrases from Phrase Alignments

Marcu and Wong [2002] proposed a translation model that assumes that lexical correspondences can be established not only at the word level, but at the phrase level as well. To learn such correspondences, they introduced a phrase-based joint probability model that simultaneously generates both the source and target sentences in a parallel corpus. Expectation Maximization learning in Marcu and Wong's framework yields both (i) a joint probability distribution $\phi(\bar{e}, \bar{f})$, which reflects the probability that phrases $\bar{e}$ and $\bar{f}$ are translation equivalents; (ii) and a joint distribution $d(i, j)$, which reflects the probability that a phrase at position i is translated into a phrase at position j. To use this model in the context of our framework, we simply marginalize the joint probabilities estimated by Marcu and Wong [2002] to conditional probabilities. Note that this approach is

Method	Training corpus size					
	10k	20k	40k	80k	160k	320k
WAIPh	84k	176k	370k	736k	1536k	3152k
Joint	125k	220k	400k	707k	1254k	2214k
Syn	19k	24k	67k	105k	217k	373k

Table 3.2: Size of the phrase translation table in terms of distinct phrase pairs (maximum phrase length 4)

consistent with the approach taken by Marcu and Wong themselves, who use conditional models during decoding.

3.3.4.4 Empirical Comparison

We used the Europarl corpus [Koehn, 2002] to carry out experiments. This corpus contains over 20 million words in each of the eleven official languages of the European Union, covering the proceedings of the European Parliament 1996-2001. 1755 sentences of length 5-15 were reserved for testing.

We translate from German to English. We measure performance using the BLEU score [Papineni et al., 2001], which estimates the accuracy of translation output with respect to a reference translation.

We compared the performance of the three methods for phrase extraction head-on, using the same decoder (Section 3.3.3) and the same trigram language model. Figure 3.10 displays the results.

In direct comparison, learning all phrases consistent with the word alignment (WAIPh) is superior to the joint model (Joint), though not by much. The restriction to only syntactic phrases (Syn) is harmful. We also included in the figure the performance of an IBM Model 4 word-based translation system (M4), which uses a greedy decoder [Germann et al., 2001]. Its performance is worse than both WAIPh and Joint. These results are consistent over training corpus sizes from 10,000 sentence pairs to 320,000 sentence pairs. All systems improve with more data.

Table 3.2 lists the number of distinct phrase translation pairs learned by each method and each corpus. The number grows almost linearly with the training corpus size, due to the large number of singletons. The syntactic restriction eliminates over 80% of all phrase pairs.

Note that the millions of phrase pairs learned fit easily into the working memory of modern computers. Even the largest models take up only a few hundred megabyte of RAM. With corpora larger than the Europarl corpus, the size of the phrase translation table does become an issue.

3.3.4.5 Weighting Syntactic Phrases

We have shown that restricting ourselves to phrases that span syntactic constituents leads to very bad machine translation performance. The restriction on syntactic phrases

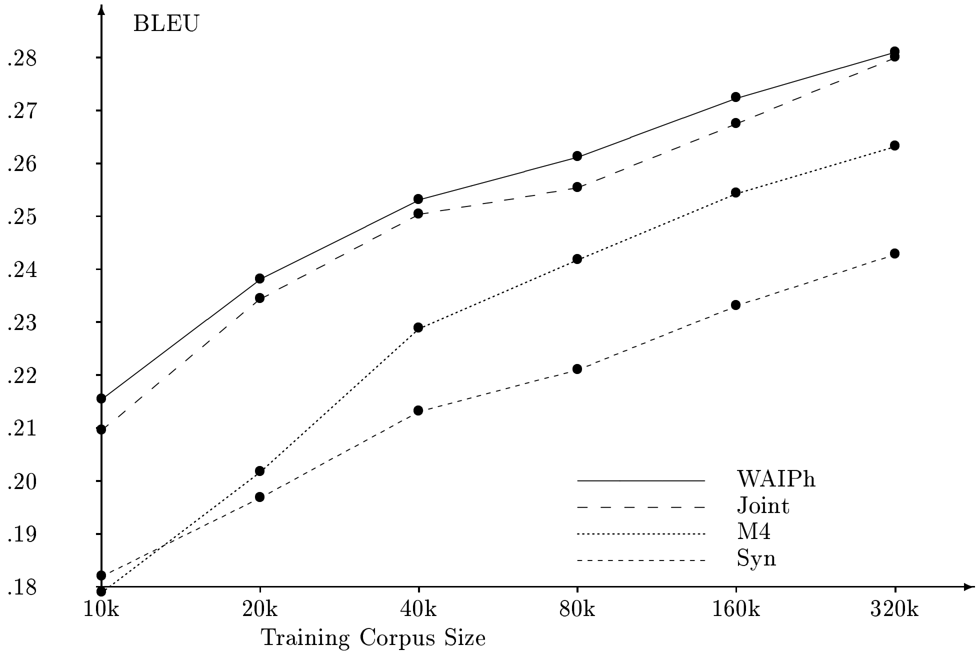


Figure 3.10: Comparison of phrase table extraction methods: all phrase pairs consistent with a word alignment (WAIPh), phrase pairs from the joint model (Joint), and only syntactic phrases (Syn). As a comparison, the word based IBM Model 4 (M4)

is harmful, because too many phrases are eliminated. But still, we might suspect, that these lead to more reliable phrase pairs.

One way to check this is to use all phrase pairs and give more weight to syntactic phrase translations. This can be done either during the data collection – say, by counting syntactic phrase pairs twice – or during translation – each time the decoder uses a syntactic phrase pair, it credits a bonus factor to the hypothesis score.

We found that neither of these methods result in significant improvement of translation performance. Even penalizing the use of syntactic phrase pairs does not harm performance significantly. These results suggest that requiring phrases to be syntactically motivated does not lead to better phrase pairs, but only to fewer phrase pairs, with the loss of a good amount of valuable knowledge.

One illustration for this is the common German *es gibt*, which literally translates as *it gives*, but really means *there is*. *Es gibt* and *there is* are not syntactic constituents. Note that also constructions such as *with regard to* and *note that* have fairly complex syntactic representations, but often simple one word translations. Allowing to learn phrase translations over such sentence fragments is important for achieving high performance.

This concludes our discussion of phrase-based machine translation for now. We us for our NP/PP translation subsystem and full sentence integration system the WAIPh method without weighting for syntactic phrases. More details on the properties of this method can be found in Appendix B.

3.4 N-Best Lists of Translation Candidates

Recall that our NP/PP translation subsystem is design as a reranking method over a set of candidate translations. This n-best list has to be generated by the base model that we described in the last section.

An n-best list is one way to represent multiple translation candidates. Such a set of possible translations can also be represented by word graphs [Ueffing et al., 2002] or forest structures over parse trees [Langkilde, 2000]. These alternative data structures allow for more compact representation of much larger set of candidates. However, it is much harder to detect and score global properties over such data structures.

Therefore, we limit ourselves to the simpler n-best lists. In the next section we will describe how we extended the decoder to produce n-best lists, and the subsequent section gives empirical evidence that a short n-best list is good enough for our purposes.

3.4.1 Generating an n-Best List

The beam search decoder we described in Section 3.3.3, in its original design, searches for a single best translation. We will now describe how we extended the decoder to generate a list of the n-best translations.

Recall the process of state expansions, illustrated in Figure 3.6. The generated hypotheses and the expansions that link them form a graph. Paths branch out when there are multiple translation options for a hypothesis from which multiple new hypotheses can be derived. Paths join when hypotheses are recombined.

3.4.1.1 Additional Arcs in the Search Graph

Usually, when we recombine hypotheses, as described in Section 3.3.3.3, we simply discard the worse hypothesis, since it cannot possibly be part of the best path through the search graph (in other words, part of the best translation).

But since we are now also interested in the second best translation, we cannot simply discard information about that hypothesis. If we would do this, the search graph would only contain one path for each hypothesis in the last hypothesis stack (which contains hypotheses that cover all foreign words).

If we store information that there are multiple ways to reach a hypothesis, the number of possible paths also multiplies along the path when we traverse backward through the graph.

In order to keep the information about merging paths, we keep a record of such merges that contains

- identifier of the previous hypothesis
- identifier of the lower-cost hypothesis
- cost from the previous to higher-cost hypothesis

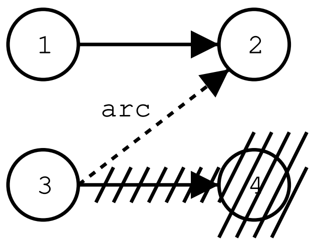


Figure 3.11: Keeping a record of an arc for n-best list generation: if hypothesis 2 and 4 are equivalent with respect to the heuristic search, hypothesis 4 is deleted (hypothesis recombination), but a record of the arc(3, 2, $\text{cost}_4 - \text{cost}_3$) is kept.

Figure 3.11 gives an example for the generation of such an arc: in this case, the hypotheses 2 and 4 are equivalent in respect to the heuristic search, as detailed in Section 3.3.3.3. Hence, hypothesis 4 is deleted. But since we want keep the information about the path leading from hypothesis 3 to 2, we store a record of this arc. The arc also contains the cost added from hypothesis 3 to 4. Note that the cost from hypothesis 1 to hypothesis 2 does not have to be stored, since it can be recomputed from the hypothesis data structures.

3.4.1.2 Mining the Search Graph for an n-Best List

The graph of the hypothesis space can be also be viewed as a probabilistic finite state automaton. The hypotheses are states, and the records of back-links and the additionally stored arcs are state transitions. The added probability scores when expanding a hypothesis are the costs of the state transitions.

Finding the n-best path in such a probabilistic finite state automaton is a well-studied problem. In our implementation, we store the information about hypotheses, hypothesis transitions, and additional arcs in a file that can be processed by the finite state toolkit Carmel[2], which we use to mine the n-best lists. This toolkit uses the n shortest paths algorithm by Eppstein [1994].

Our method is related to work by Ueffing et al. [2002] for generating n-best lists for IBM Model 4.

3.4.2 Acceptable Translations in n-Best List

One key question for our approach is how often an acceptable translation can be found in an n-best list. For a test corpus of 1362 NP/PPs, we generated n-best lists of size up to 100, and checked manually if an acceptable translation can be found in list of size n. The answer to this is illustrated in Figure 3.12.

[2]available at `http://www.isi.edu/licensed-sw/carmel/`

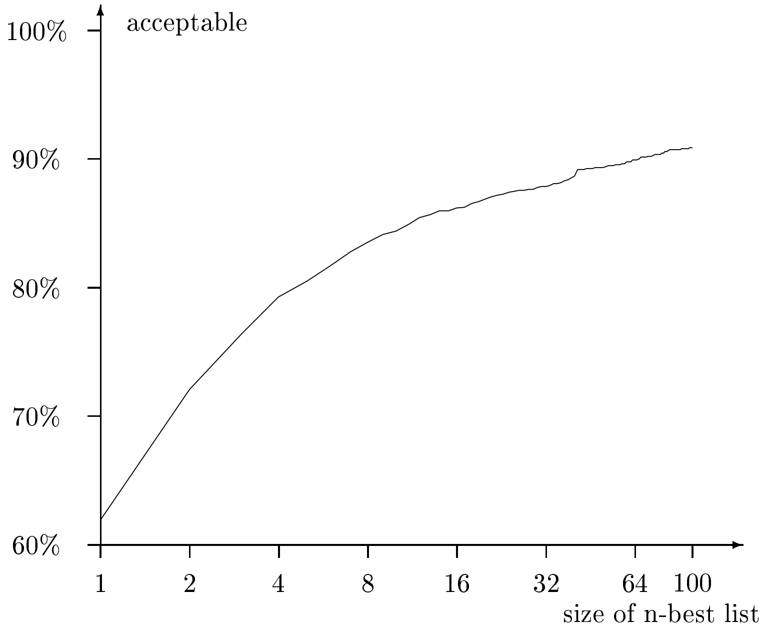


Figure 3.12: Ratio of NP/PPs for which an acceptable NP/PP translations can be found in n-best list of candidate translations for different sizes n

While an acceptable translation comes out on top for only about 60% of the NP/PPs in our test corpus, one can be found in the 100-best list for over 90% of the NP/PPs[3]. This means that rescoring has the potential to raise performance by 30%.

3.5 Maximum Entropy Reranking

Given an n-best list of candidates and additional features, we transform the translation task from a search problem into a reranking problem: instead of creating and searching a hypothesis space of possible translations for the best translation, we now have to pick the best one among of 100 translations.

The main motivation for this approach is that we now have the full translation available and that we can define global features over the given pair of foreign input and English translation.

3.5.1 Overview

Each translation candidate in the n-best list is represented by features value for a predefined set of features. We describe the features in detail in Chapter 4. The initial features are the logarithm of the probability scores that the model assigns to each candidate

[3]Note that these numbers are obtained after compound splitting, described in Section 4.1.

translation: the language model score, the phrase translation score, and the reordering (distortion) score.

The feature values are combined by a log-linear model

$$p_\lambda(e|f) = \exp \sum_i \lambda_i h_i(f, e) \tag{3.2}$$

where e is a candidate translation, f is the foreign input, the h_i's are the feature values and the λ_i's are the feature weights.

The task of the maximum entropy learner is to find values for the feature weights λ_i. This is done using a development corpus of NP/PPs, for which feature values and acceptable translation in the n-best list are known. The next section describes the development corpus, the subsequent section the mathematics behind maximum entropy training.

3.5.2 Development Corpus

The development corpus is taken from the same Europarl corpus that was used for training the baseline machine translation system and whose acquisition is described in Section 3.2. However, the part of the corpus used as development corpus for maximum entropy training has not been used for training the base model.

The development corpus contains 683 foreign NP/PPs and a list of up to 100 candidate translations for each. For each candidate translation we have to generate feature values. We also need accuracy judgments for the translations, so the learner can find a set of feature weights that lead to acceptable translations of the development corpus. Multiple (or none) candidate translations can be marked as "acceptable".

3.5.3 Mathematics of Maximum Entropy Reranking

The task for the learning method is to find a probability distribution $p(e|f)$ that indicates whether the candidate translation e is an accurate translation of the input f. The decision rule to pick the best translation is

$$e_{\text{best}} = \text{argmax}_e p(e|f) \tag{3.3}$$

The development corpus provides the empirical probability distribution by distributing the probability mass over the acceptable translations $\{e_{a_i}\}$:

$$\tilde{p}(e_{a_i}|f) = |\{e_{a_i}\}|^{-1} \tag{3.4}$$

If none of the candidate translations for a given input f is acceptable, we pick the candidates that are closest to reference translations measured by minimum edit distance.

We use a maximum entropy framework to parametrize the probability distribution as

$$p_\lambda(e|f) = \exp \sum_i \lambda_i h_i(f, e) \tag{3.5}$$

where the h_i's are the feature values and the λ_i's are the feature weights.

Since we have only a sample of the possible translations e for the given input f, we normalize the probability distribution so that

$$\sum_i p_\lambda(e_i|f) = 1 \tag{3.6}$$

for our sample $\{e_i\}$ of candidate translations.

Maximum entropy learning finds a set of feature weights λ_i so that $E_{p_\lambda}[h_i] = E_{\tilde{p}}[h_i]$ for each feature h_i. These expectations are computed as sums over all candidate translations e for all inputs f:

$$\sum_{(f,e)} \tilde{p}(f) p_\lambda(e|f) h_i(f,e) = \sum_{(f,e)} \tilde{p}(f) \tilde{p}(e|f) h_i(f,e) \tag{3.7}$$

A nice property of maximum entropy training is that it converges to a global optimum. There are a number of methods and tools available to carry out this training of feature weights. We use the toolkit[4] developed by Malouf [2002]. Berger et al. [1996] and Manning and Schütze [1999] provide good introductions to maximum entropy learning.

Note that any other machine learning method, such as support vector machines, could be used as well. We chose maximum entropy for its ability to deal with both real-valued and binary features. This method is also similar to work by Och and Ney [2002], who use maximum entropy to tune model parameters.

3.5.4 Design Details

When experimenting with the maximum entropy trainer, we learned a number of small lessons. We simply report on them here, but do not claim to have scientifically investigated them by thorough experimentation.

- The n-best lists generated by the decoder contain duplicate translations. For instance, the same translation could be obtained by translating two foreign words as a phrase, or translating them as single words. It is important to remove duplicates from the n-best list. We do this by keeping only the best-scoring translation, as scored by the decoder.

- As noted earlier, for some NP/PPs, no translation in the list is judged acceptable. For our data, this is the case for about 10% of the NP/PPs. For these, the "closest acceptable translation(s)" are treated as acceptable in purposes of maximum entropy training. The performance of the trainer is sensitive to the exact method used here. We use minimum edit distance (or "word error rate"). Alternate attempts to use error measures using n-gram precision and recall led to worse results.

- Outliers in feature values can cause problems. In our initial design, the decoder produced improbably low language model scores for some translation (due to bugs in language model code related to unknown words). If some translation with extreme

[4] Available at `http://www-rohan.sdsu.edu/~malouf/pubs.html`

feature values are labeled “acceptable” and given empirical probability mass, this can dramatically hurt the weight estimation for this feature.

Chapter 4

Properties of NP/PP Translation

We will now discuss the linguistic properties of NP/PP translation that we exploit in order to improve our NP/PP translation subsystem. The first of these (compounding of words) is addressed by preprocessing, while the others motivate features that are used in n-best list reranking.

The types of properties we exploit can also be classified into two categories: Generally speaking, the performance of statistical machine translation systems can be improved by better translation modeling (which ensures correspondence between input and output) and better language modeling (which ensures fluent output). Language modeling can in general be improved by different types of language models (e.g., syntactic language models), or additional training data for the language model.

Some of the properties we exploit here are focused on the translation model (e.g., compound splitting) and others on the language model (e.g., web n-grams).

4.1 Compound Splitting

The work described in this section has been published in the paper "Empirical Methods for Compound Splitting" [Koehn and Knight, 2003].

Compounding of words is common in a number of languages (German, Dutch, Finnish, Greek, etc.). Since words may be joined freely, this vastly increases the vocabulary size, leading to sparse data problems. This poses challenges for a number of NLP applications such as machine translation, speech recognition, text classification, information extraction, or information retrieval.

For machine translation, the splitting of an unknown compound into its parts enables the translation of the compound by the translation of its parts. Take the word *Aktionsplan* in German (see Figure 4.1), which was created by joining the words *Aktion* and *Plan*. Breaking up this compound would assist the translation into English as *action plan*.

Compound splitting is a well defined computational linguistics task. One way to define the goal of compound splitting is to break up foreign words so that a one-to-one correspondence to English can be established. Note that we are looking for a one-to-one correspondence to English content words. Say that the preferred translation of *Aktionsplan* is *plan for action*. The lack of correspondence for the English word *for* does not detract from the definition of the task: We would still like to break up the German

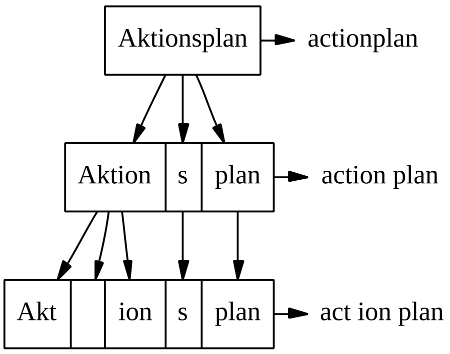


Figure 4.1: Splitting options for the German word *Aktionsplan*

compound into the two parts *Aktion* and *Plan*. The insertion of function words is not our concern in this section.

Ultimately, the purpose of this work is to improve the quality of machine translation systems. For instance, phrase-based translation systems may recover more easily from splitting regimes that do not create a one-to-one translation correspondence. One splitting method may mistakenly break up the word *Aktionsplan* into the three words *Akt*, *Ion*, and *Plan* (English glosses: *act, ion, plan*). But if we consistently break up the word *Aktion* into *Akt* and *Ion* in our training data, such a system will likely learn the translation of the word pair *Akt Ion* into the single English word *action*.

These considerations lead us to three different objectives and therefore three different evaluation metrics for the task of compound splitting:

- One-to-One correspondence
- Translation quality with a word-based translation system
- Translation quality with a phrase-based translation system

For the first objective, we compare the output of our methods to a manually created gold standard. For the second and third, we provide differently prepared training corpora to statistical machine translation systems.

4.1.1 Related Work

While the linguistic properties of compounds are widely studied [Langer, 1998], there has been only limited work on empirical methods to split up compounds for specific applications.

Brown [2002] proposes an approach guided by a parallel corpus. It is limited to breaking compounds into cognates and words found in a translation lexicon. This lexicon may also be acquired by training a statistical machine translation system. The methods leads to improved text coverage of an example based machine translation system, but no results on translation performance are reported.

Monz and de Rijke [2001] and Hedlund et al. [2001] successfully use lexicon-based approaches to compound splitting for information retrieval. Compounds are broken into either the smallest or the biggest words that can be found in a given lexicon.

Larson et al. [2000] propose a data-driven method that combines compound splitting and word recombination for speech recognition. While it reduces the number of out-of-vocabulary words, it does not improve speech recognition accuracy.

Morphological analyzers such as Morphix [Finkler and Neumann, 1998] usually provide a variety of splitting options and leave it to the subsequent application to pick the best choice.

4.1.2 Splitting Options

Compounds are created by joining existing words together. Thus, to enumerate all possible splittings of a compound, we consider all splits into known words. Known words are words that exist in a training corpus, in our case the European parliament proceedings consisting of 20 million words of German [Koehn, 2002].

When joining words, filler letters may be inserted at the joint. (recall the example of *Aktionsplan*, where the letter *s* was inserted between *Aktion* and *Plan*). Rather than try to implement rules for how and when such letters may be inserted, we allow them between any two words. As fillers we allow *s* and *es* when splitting German words, which covers almost all cases. Other transformations at joints include dropping of letters, such as when *Schweigen* and *Minute* are joined into *Schweigeminute*, dropping an *n*. A extensive study of such transformations is carried out by Langer [1998] for German.

To summarize: We try to cover the entire length of the compound with known words and fillers between words. An algorithm to break up words in such a manner could be implemented using dynamic programming, but since computational complexity is not a problem, we employ an exhaustive recursive search. To speed up word matching, we store the known words in a hash based on the first three letters. Also, we restrict known words to words of at least length three.

For the word *Aktionsplan*, we find the following splitting options:

- aktionsplan
- aktion–plan
- aktions–plan
- akt–ion–plan

We arrive at these splitting options, since all the parts – *aktionsplan, aktions, aktion, akt, ion,* and *plan* – have been observed as whole words in the training corpus.

These splitting options are the basis of our work. In the following we discuss methods that pick one of them as the correct splitting of the compound:

- frequency-based
- guided by a parallel corpus
- using part-of-speech information

4.1.3 Frequency Based Metric

The more frequently a word occurs in a training corpus, the bigger the statistical basis to estimate translation probabilities, and the more likely the correct translation probability distribution is learned [Koehn and Knight, 2001]. This insight leads us to define a splitting metric based on word frequency.

Given the count of words in the corpus, we pick the split S with the highest geometric mean of word frequencies of its parts p_i (n being the number of parts):

$$\text{argmax}_{\text{S}}\Big(\prod_{p_i \in S} \text{count}(p_i)\Big)^{\frac{1}{n}} \tag{4.1}$$

Since this metric is purely defined in terms of German word frequencies, there is not necessarily a relationship between the selected option and correspondence to English words. If a compound occurs more frequently in the text than its parts, this metric would leave the compound unbroken – even if it is translated in parts into English.

In fact, this is the case for the example *Aktionsplan*. Again, the four options:

- aktionsplan(852) $\rightarrow$ 852
- aktion(960)–plan(710) $\rightarrow$ 825.6
- aktions(5)–plan(710) $\rightarrow$ 59.6
- akt(224)–ion(1)–plan(710) $\rightarrow$ 54.2

Behind each part, we indicated its frequency in parenthesis. On the right side is the geometric mean score of these frequencies. The score for the unbroken compound (852) is higher than the preferred choice (825.6).

On the other hand, a word that has a simple one-to-one correspondence to English may be broken into parts that bear little relation to its meaning. We can illustrate this on the example of *Freitag* (English: *Friday*), which is broken into *frei* (English: *free*) and *Tag* (English: *day*):

- frei(885)–tag(1864) $\rightarrow$ 1284.4
- freitag(556) $\rightarrow$ 556

As we can see, this method makes mistakes, which we try to address with the following improved methods.

4.1.4 Guidance from a Parallel Corpus

As stated earlier, one of our objectives is the splitting of compounds into parts that have one-to-one correspondence to English. One source of information about word correspondence is a parallel corpus: text in a foreign language, accompanied by translations into English. Usually, such a corpus is provided in form of sentence translation pairs.

Going through such a corpus, we can check for each splitting option if its parts have translations in the English translation of the sentence. In the case of *Aktionsplan* we

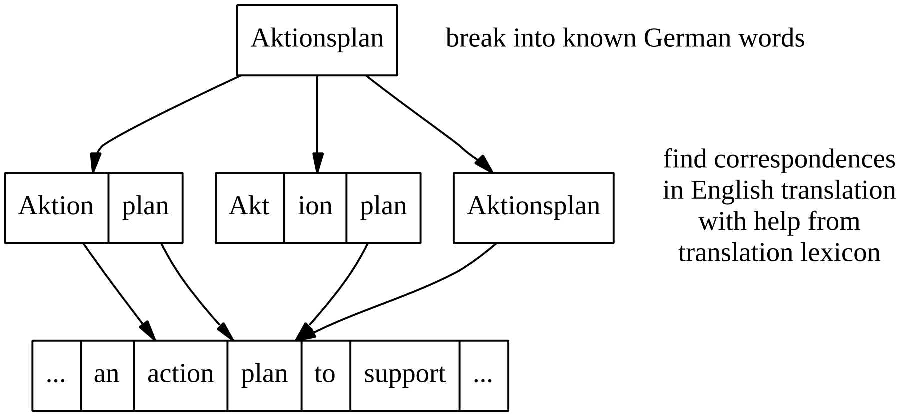


Figure 4.2: Acquisition of splitting knowledge from a parallel corpus: The split *Aktion–plan* is preferred since it has most coverage with the English (two words overlap).

would expect the words *action* and *plan* on the English side, but in case of *Freitag* we would not expect the words *free* and *day*. This would lead us to break up *Aktionsplan*, but not *Freitag*. See Figure 4.2 for illustration of this method.

This approach requires a translation lexicon. The easiest way to obtain a translation lexicon is to learn it from a parallel corpus. This can be done with the toolkit Giza [Al-Onaizan et al., 1999], which establishes word-alignments for the sentences in the two languages.

With this translation lexicon we can perform the method alluded to above: For each German word, we consider all splitting options. For each splitting option, we check if it has translations on the English side.

To deal with noise in the translation table, we demand that the translation probability of the English word given the German word be at least 0.01. We also allow each English word to be considered only once: If it is taken as evidence for correspondence to the first part of the compound, it is excluded as evidence for the other parts. If multiple options match the English, we select the one(s) with the most splits and use word frequencies as the ultimate tie-breaker.

Second Translation Table

While this method works well for the examples *Aktionsplan* and *Freitag*, it failed in our experiments for words such as *Grundrechte* (English: *basic rights*). This word should be broken into the two parts *Grund* and *Rechte*. However, *Grund* translates usually as *reason* or *foundation*. But here we are looking for a translation into the adjective *basic* or *fundamental*. Such a translation only occurs when *Grund* is used as the first part of a compound.

To account for this, we build a second translation lexicon as follows: First, we break up German words in the parallel corpus with the frequency method. Then, we train a translation lexicon using Giza from the parallel corpus with split German and unchanged English.

Since in this corpus *Grund* is often broken off from a compound, we learn the translation table entry *Grund*$\leftrightarrow$*basic*. By joining the two translation lexicons, we can apply the same method, but this time we correctly split *Grundrechte*.

By splitting all the words on the German side of the parallel corpus, we acquire a vast amount of splitting knowledge (for our data, this covers 75,055 different words). This knowledge contains for instance, that *Grundrechte* was split up 213 times, and kept together 17 times.

When making splitting decisions for new texts, we follow the most frequent option based on the splitting knowledge. If the word has not been seen before, we use the frequency method as a back-off.

4.1.5 Limitation on Part-Of-Speech

A typical error of the methods presented so far is that prefixes and suffixes are often split off. For instance, the word *folgenden* (English: *following*) is broken off into *folgen* (English: *consequences*) and *den* (English: *the*). While this is nonsensical, it is easy to explain: The word *the* is commonly found in English sentences, and therefore taken as evidence for the existence of a translation for *den*.

Another example for this is the word *Voraussetzung* (English: *condition*), which is split into *vor* and *aussetzung*. The word *vor* translates to many different prepositions that frequently occur in English.

To exclude these mistakes, we use information about the parts-of-speech of words. We do not want to break up a compound into parts that are prepositions or determiners, but only content words: nouns, adverbs, adjectives, and verbs.

To accomplish this, we tag the German corpus with POS tags using the TnT tagger [Brants, 2000]. We then obtain statistics on the parts-of-speech of words in the corpus. This allows us to exclude words based on their POS as possible parts of compounds. We limit possible parts of compounds to words that occur most of the time as one of following POS: ADJA, ADJD, ADV, NN, NE, PTKNEG, VVFIN, VVIMP, VVINF, VVIZU, VVPP, VAFIN, VAIMP, VAINF, VAPP, VMFIN, VMINF, VMPP.

4.1.6 Evaluation

The training set for the experiments is a corpus of 650,000 noun phrases and prepositional phrases (NP/PP). For each German NP/PP, we have an English translation. This data was extracted from the Europarl corpus [Koehn, 2002], with the help of a German and English statistical parser, as described in the previous chapter (Section 3.2). This limitation is purely for computational reasons, since we expect most compounds to be nouns. An evaluation of full sentences is expected to show similar results.

We evaluate the performance of the described methods on a blind test set of 1000 NP/PPs, which contain 3498 words. Following good engineering practice, the methods have been developed with a different development test set. This restrains us from overfitting to a specific test set.

Method	Correct		Wrong			Metrics		
	split	**not**	**not**	**faulty**	**split**	**prec.**	**recall**	**acc.**
raw	0	3296	202	0	0	-	0.0%	94.2%
eager	148	2901	3	51	397	24.8%	73.3%	87.1%
frequency based	175	3176	19	8	122	57.4%	86.6%	95.7%
using parallel	180	3270	13	9	27	83.3%	89.1%	98.6%
using parallel and POS	182	3287	18	2	10	93.8%	90.1%	**99.1%**

Table 4.1: Evaluation of the methods compared against a manually annotated gold standard of splits: using knowledge from parallel corpus and part-of-speech information gives the best accuracy (99.1%).

4.1.6.1 One-to-one Correspondence

Recall that our first objective is to break up German words into parts that have a one-to-one translation correspondence to English words. To judge this, we manually annotated the test set with correct splits. Given this gold standard, we can evaluate the splits proposed by the methods.

The results of this evaluation are given in Table 4.1. The columns in this table mean:

correct split: words that should be split and were split correctly

correct non: words that should not be split and were not

wrong not: words that should be split but were not

wrong faulty split: words that should be split, were split, but wrongly (either too much or too little)

wrong split: words that should not be split, but were

precision: (correct split) / (correct split + wrong faulty split + wrong superfluous split)

recall: (correct split) / (correct split + wrong faulty split + wrong not split)

accuracy: (correct) / (correct + wrong)

To briefly review the methods:

raw: unprocessed data with no splits

eager: biggest split, i.e., the split into as many parts as possible. If multiple biggest splits are possible, the one with the highest frequency score is taken.

frequency based: split into most frequent words, as described in Section 4.1.3

using parallel: split guided by splitting knowledge from a parallel corpus, as described in Section 4.1.4

Method	BLEU
raw	0.291
eager	0.222
frequency based	**0.317**
using parallel	0.294
using parallel and POS	0.306

Table 4.2: Evaluation of the methods with a word based statistical machine translation system (IBM Model 4). Frequency based splitting is best, the methods using splitting knowledge from a parallel corpus also improve over unsplit (raw) data.

using parallel and POS: as previous, with an additional restriction on the POS of split parts, as described in Section 4.1.5

Since we developed our methods to improve on this metric, it comes as no surprise that the most sophisticated method that employs splitting knowledge from a parallel corpus and information about POS tags proves to be superior with 99.1% accuracy. Its main remaining source of error is the lack of training data. For instance, it fails on more obscure words such as *Passagier–aufkommen* (English: *passenger volume*), where even some of the parts have not been seen in the training corpus.

4.1.6.2 Translation Quality with Word Based Machine Translation

The immediate purpose of our work is to improve the performance of statistical machine translation systems. Hence, we use the splitting methods to prepare training and testing data to optimize the performance of such systems.

First, we measured the impact on a word based statistical machine translation system, the widely studied IBM Model 4 [Brown et al., 1990], for which training tools [Al-Onaizan et al., 1999] and decoders [Germann et al., 2001] are freely available. We trained the system on the 650,000 NP/PPs with the Giza toolkit, and evaluated the translation quality on the same 1000 NP/PP test set as in the previous section. Training and testing data was split consistently in the same way. The translation accuracy is measured against reference translations using the BLEU score [Papineni et al., 2002]. Table 4.2 displays the results.

Somewhat surprisingly, the frequency based method leads to better translation quality than the more accurate methods that take advantage from knowledge from the parallel corpus. One reason for this is that the system recovers more easily from words that are split too much than from words that are not split up sufficiently. Of course, this has limitations: Eager splitting into as many parts as possible fares abysmally.

4.1.6.3 Translation Quality with Phrase Based Machine Translation

Compound words violate the bias for one-to-one word correspondences of word based SMT systems. This is one of the motivations for phrase based systems that translate groups

Method	BLEU
raw	0.305
eager	**0.344**
frequency based	**0.342**
using parallel	0.330
using parallel and POS	0.326

Table 4.3: Evaluation of the methods with a phrase based statistical machine translation system. The ability to group split words into phrases overcomes the many mistakes of maximal (eager) splitting of words and outperforms the more accurate methods.

of words. One of such systems is the joint model proposed by Marcu and Wong [2002]. We trained this system with the different flavors of our training data, and evaluated the performance as before. Table 4.3 shows the results.

Here, the eager splitting method that performed so poorly with the word based SMT system comes out ahead. The task of deciding the granularity of good splits is deferred to the phrase based SMT system, which uses a statistical method to group phrases and rejoin split words. This turns out to be even slightly better than the frequency based method.

4.1.7 Conclusion

We introduced various methods to split compound words into parts. Our experimental results demonstrate that what constitutes the optimal splitting depends on the intended application. While one of our methods reached 99.1% accuracy compared against a gold standard of one-to-one correspondences to English, other methods showed superior results in the context of statistical machine translation. For this application, we could improve the translation quality by up to 0.039 points as measured by the BLEU score.

The words resulting from compound splitting could also be marked as such, and not just treated as regular words, as they are now. Future machine translation models that are sensitive to such linguistic clues might benefit even more.

In the evaluation of NP/PP translation accuracy, on which report in the conclusion of this Chapter, compound splitting improved performance by +2.8%. In other words, 2.8% more NP/PPs could be translated correctly after compound splitting.

4.2 Web n-Grams

Using more data is almost always a safe bet in a computational linguistics application. This makes the world wide web an attractive resource. The search engine Google[1] indexes 3 billion web pages, with an estimated word count of maybe a trillion. Compare that to

[1] `http://www.google.com/`

	16m word corpus		2b document web	
	count	ratio	count	ratio
full	96 of 164	58%	124 of 164	75%
2-gram	489 of 525	93%	523 of 525	99%
3-gram	271 of 374	72%	363 of 374	97%
4-gram	113 of 264	42%	225 of 264	85%
5-gram	49 of 190	25%	124 of 190	65%
6-gram	21 of 133	15%	59 of 133	44%
7-gram	15 of 91	16%	28 of 91	30%

Table 4.4: n-Gram existence on the web compared to a large training corpus

the largest available text corpora (around a billion words) and typical training corpora for statistical machine translation (10–100 million words).

4.2.1 n-Gram Existence on the Web

Having 1000 times more English text is a powerful resource for language modeling. To shed some more light on this, we carried out a study that checked how often we can find an n-gram of new text in a corpus from the same domain and on the web.

The results are displayed in Table 4.4. We investigated how many of the correct English translations of a 164 German-English NP/PP corpus (the same corpus mentioned in Section 1.6) can be found on the web, compared to how many can be found in the English side of the parallel corpus. We did the same for n-grams of varying length.

The results suggest that a NP/PP translation system should hardly ever output a bigram or even trigram that has not been seen on the web.

But also larger n-grams can be frequently seen on the web. 85% (vs. 42% in the training corpus) of the 4-grams can be found, 65% (vs. 25% in the training corpus) of the 5-grams can be found. A feature of the form "all 5-grams in the candidate translation have been seen on the web" may provide important support for certain candidate translations.

We argue that a NP/PP translation can be improved by biasing it towards output that contains frequent n-grams. This is already done in trigram language models which have a bias for frequent bigrams and trigrams. What is the impact of this bias?

To assess this, we carried out the same experiment on output from a machine translation system that uses a trigram language model trained on the 16 million word training corpus. The results are summarized in Table 4.5.

Note the effect of the trigram language model: The system output contains known bigrams, trigrams and even quadgrams from the training corpus with a similar frequency as the correct translations: 91% vs. 93% for bigrams, 73% vs. 72% for trigrams, 43% vs. 42% for quadgrams. This is not the case for larger n-grams (4% vs. 16% for 7-grams), and for n-grams seen on the web.

	16m word corpus		2b document web	
	system	correct	system	correct
full	53%	58%	66%	75%
2-gram	91%	93%	97%	99%
3-gram	73%	72%	92%	97%
4-gram	43%	42%	80%	85%
5-gram	20%	25%	56%	65%
6-gram	5%	15%	32%	44%
7-gram	4%	16%	14%	30%

Table 4.5: n-Gram existence on the web compared to a large training corpus, both for machine translation system output and reference NP/PP

4.2.2 n-Gram Existence and Frequency as Features

There are various ways one may integrate this vast resource into a machine translation system. For our candidate translations for noun phrases, we collected the frequency of all their n-grams with n up to 7, and the frequency of the whole phrase with a special tool in cooperation with the Google search engine.

Given these frequencies, we considered the following feature types:

- **Building a n-gram language model:** Given the n-gram frequencies, we can build a classic n-gram language model that predicts the likelihood of a sequence of words given their preceding words:

  $$p(s) = p(w_0, ..., w_N) = \prod_{n=0}^{N} p(w_n|w_0, ..., w_{n-1}) \tag{4.2}$$

  The probabilities for each word are computed by a back-off formula

  $$p(w_n|w_0, ..., w_{n-1}) = \sum_{i=0}^{n-1} \lambda_{n-i} p_{n-i}(w_n|w_i, ..., w_{n-1}) \tag{4.3}$$

  where the n-gram probabilities are computed by maximum likelihood

  $$p_{n-i}(w_n|w_{n-i}, ..., w_{n-1}) = \frac{\text{count}(w_{n-i}, ..., w_n)}{\sum_{w_n} \text{count}(w_{n-i}, ..., w_n)} \tag{4.4}$$

  Usually some smoothing is applied to the n-gram probability distributions. The λ's are obtained by using the EM algorithm on held out data.

  Doing this for web frequencies poses a number of challenges: Since we have only limited access to the data, smoothing and parameter estimation is very hard. For more on language modeling refer to an overview by Chen and Goodman [1998].

- **Existence on the web:** A much simper way to featurize the web counts is to introduce a feature for each n that checks if all the n-grams of size n in the candidate translation occur on the web or not.
- **Frequency on the web:** Instead of creating the binary features above, we can use as a feature value the log-frequency of the n-Gram on the web. While this is straight-forward for the frequency of the entire phrase, it is more tricky for, say, bigrams. Should we use the sum, the max, the min, or some form of average of all bigram frequencies as feature value?
- **Derivatives of frequencies:** Somewhat between using raw frequencies and simple existence, we can also devise features that check if all n-grams of size n occur at least m times on the web.

Note that for binary features, we can choose to use positive features (do all bigrams exist on the web?), negative features (do some bigrams not exist on the web?), or both.

What should be the feature value for the binary feature "do all trigrams in the candidate translation occur on the web?" for a two word candidate translation? Since all trigrams in a candidate translation almost always occur on the web (recall the above 97%), not having a positive value for the trigram feature puts the two-word candidate translation at a disadvantage.

4.2.3 Experiments

Through experimentation, we settled on the following set of features:

- Does the candidate translation as a whole occur on the web?
- Do all n-grams in the candidate translation occur on the web?
- Do all n-grams in the candidate translation occur at least 10 times on the web?

We use both positive and negative features for n-grams of the size 2 to 7. Using web count features gives us an improvement of +2.2% on NP/PP translation accuracy in the experiment reported in the end of this chapter.

4.3 Syntactic Features

Unlike in decoding, for reranking we have the complete candidate translation available. This means that we can define features that address any property of the full NP/PP translation pair. One such set of features is syntactic features.

4.3.1 Syntactic Alignment of NP/PPs

By syntactic features we mean features that are defined using information of the syntactic parse trees of the foreign input NP/PP and the English output NP/PP. The syntactic parse tree for the foreign NP/PP is available before decoding: We had to parse the foreign sentence in order to detect the NP/PP.

Since our base translation model — phrase-based translation — does not pay attention to syntax, the output English candidate NP/PP translations do not contain any syntactic information. We have to add this syntactic markup after the translation is produced.

This is done in two steps: first part-of-speech tagging, then syntactic parsing. For both these tasks, standard tools are readily available. As for the original corpus preparation (see Section 3.2), we used Brill [1995]'s transformation-based tagger and Collins [1997]'s statistical syntactic parser.

However, since we are parsing NP/PP that are output of a statistical machine translation system, this required some small modifications. First, for machine translation characters are typically lowercased (to reduce unnecessary variations such as *the*, *The*, and *THE*). Secondly, we know that we do not have any verbs in the English output NP/PP, but the POS tagger might still want to tag some words as NP/PP. This led us to retrain the POS tagger on the English side of the parallel NP/PP corpus, reducing the error rate significantly. We did not modify the parser.

To align the given foreign syntax tree and the generated English NP/PP syntax tree, we also need a word alignment between the two. However, the basic phrase-based translation model translates sequences of words, instead of individual words. Yet, since we learn the phrase translations from a word alignment, we can remember the word alignment for each phrase pair and recall it for a given phrase translation application.

Figure 4.3 illustrates the steps that give us a word-aligned pair of syntax trees. We can now use any computable property over the pair of trees as a feature.

4.3.2 Preservation of the Number of a Noun

Generally, plural nouns translate as plural nouns, while singular nouns translate as singular. Of course, it is easy to find exceptions to this, but picking the wrong number for a noun is a common mistake in statistical machine translation. Giving the system a bias for preservation of the number of noun should be overall beneficial.

We check every alignment between nouns for number preservation, using the word alignment and the part-of-speech tags. If a noun is aligned to multiple nouns, we only consider the last noun in the noun group (e.g. *state secrets* is plural, *Staatsgeheimnisse* is as well).

This feature illustrates some of the design motivations of introducing syntax into NP/PP translation. The features encode relevant general syntactic knowledge about the translation of noun phrases. We define soft constraints that may be overruled by other components of the system. We shy away from hard rules, since there are always exceptions and, also, errors in the tools that generate the syntactic markup.

4.3.3 Preservation of Prepositions

The phrase-based translation model is oblivious to the role of prepositions. It may drop them, if it has seen often alignments between specific NP/PP pairs, where only one has a leading preposition, say, due to different subcategorization of the verb.

However, prepositions are generally translated between English and German, with the exception of (1) German genitive noun phrase that typically translate as prepositional

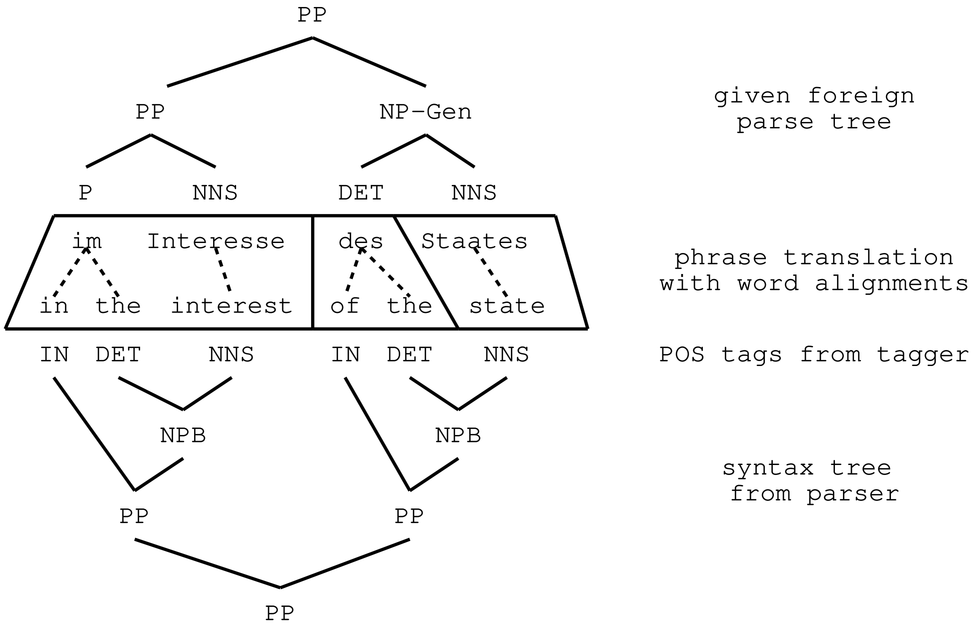


Figure 4.3: Generation of a word-aligned pair of syntax tree: The foreign syntax tree is given, phrase translation also provides the word alignment, the English syntax tree is obtained using a POS tagger and syntactic parser.

phrases, (2) dropping and adding of prepositions due to reordering (*state interest* becoming *interest of the state*), (3) dropping and adding of leading prepositions due to different attachment to the clause level.

We check for preservation of all prepositions, except for the leading preposition of the NP/PPs. German genitive noun phrases are treated as prepositional phrases with a preposition. We allow dropping of prepositions during movement.

4.3.4 Number Agreement in BaseNP

The first two features we described are translation model features. In other words, they introduce a bias for syntactically correct transformations. But we can also use syntax to overcome weaknesses of the language model.

Consider the noun phrase *this nice green flowers*. A trigram language model will not be able to detect the violation of number agreement between the singular *this* and the plural *flowers*, since there are two words in-between. Even if there are only one or no words between words that have to be in agreement, the trigram language model may not have seen that particular example to make the right decision.

We introduce number agreement in baseNPs as a syntactic feature. We check for every noun phrase, if the determiner agrees in number with the final noun (*a*, *an*, *this*, *that*, *every* and *each* indicate singular, *those*, *these*, *some*, and *a few* indicate plural).

4.3.5 Design of Features

So far, we described the features as computable properties that may be correct or violated in a given pair of syntax trees. To use them as features for maximum entropy reranking, we can realize them as integers that count how many times a particular properties has been violated and how many times it is correct. There is also the third case, that the property does not apply to a given example. Consider one NP/PP translation with one baseNP, and an alternative NP/PP translation with two baseNPs. In the second case, the number agreement in baseNPs may be true twice, but there is no motivation to reward this.

Because of these, we use the syntactic features as negative features that flag violations. Multiple violations of the same property in a given NP/PP could occur, so the feature value may be bigger than 1.

Syntactic features give us +0.7% accuracy in NP/PP translation in the experiments in the next section.

4.4 Experiments

We described in this chapter the properties of NP/PP translation that we exploit to improve translation quality. We will now describe the experiment we carried out to evaluate this.

System	NP/PP Correct	
IBM Model 4	734	53.9%
Phrase Model	814	59.8%
Compound Splitting	853	62.6%
Re-Estimated Parameters.	874	64.2%
Web Count Features	904	66.4%
Syntactic Features	914	67.1%

Table 4.6: Improving noun phrase translation with special modeling and additional features: Correct NP/PPs and BLEU score for overall sentence translation

4.4.1 Quantitative Evaluation

We evaluate the performance of our NP/PP translation subsystem on a blind test set of 1362 NP/PPs extracted from 534 sentences. This test corpus was created in the same way as the development corpus (see Section 3.5.2). It does not overlap with the training corpus for the baseline system and the development corpus for the maximum entropy learner that sets the feature weights.

In order to show the contributions of the different additions to the system, we evaluated the performance at each stage. We use the same accuracy judgments used for the development corpus. The results are displayed in Table 4.6.

Starting from the IBM Model 4 baseline, we achieve gains using our phrase-based translation model (+5.9%), applying compound splitting to training and test data (+2.8%), re-estimating the weights for the system components using the maximum entropy reranking framework (+1.6%), adding web count features (+2.2%) and syntactic features (+0.7%). Overall we achieve an improvement of 13.2% over the baseline. Improvements of 2.5% are statistically significant given the size of our test corpus. For more on statistical significance, please also refer to Appendix A.

4.4.2 Discussion

We have shown that noun phrase translation can be separated out as a subtask. Our manual experiments show that NP/PPs can almost always be translated as NP/PPs across languages, and that the translation of NP/PPs usually does not require additional external context.

We also demonstrated that the reduced complexity of noun phrase translation allows us to address the problem in a maximum entropy reranking framework, where we only consider the 100-best candidates of a base translation system. This enables us to introduce any features that can be computed over a full translation pair, instead of being limited to features that can be integrated into the search algorithm of the decoder, which only has access to partial translations.

Error	Frequency
Unknown Word	34%
Tagging or parsing error	28%
Unknown translation	14%
Complex syntactic restructuring	7%
Too long	6%
Could not drop leading preposition	3%
Untranslatable	2%
Other	6%

Table 4.7: Error analysis for NP/PPs without acceptable translation in 100-best list

We improved performance of noun phrase translation by 13.2% by using a phrase translation model, a maximum entropy reranking method and addressing specific properties of noun phrase translation: compound splitting, using the web as a language model, and syntactic features.

In the next chapter, Chapter 5, we will describe how the NP/PP translation module can be integrated into a full sentence translation system.

4.5 Error Analysis

In order to better understand the deficiencies of our system, we carried out two studies to examine why our NP/PP translation module fail on some NP/PPs. Translation may fail because our n-best list does not contain an acceptable translation at all (this is the case for 10% of the NP/PPs), or because our re-ranker does not pick an acceptable translation out of the list (this is the case for 25% of the NP/PPs).

4.5.1 No Acceptable Translation in n-Best List

What are the problems with the 10% of NP/PPs for which no translation can be found? To investigate this, we carried out an error analysis of these NP/PPs. Results are given in Table 4.7.

Almost half of the errors can be attributed to unknown words or unknown translations (48%), but also parsing or tagging errors are a big issue (28%).

The causes in detail are:

- **Unknown Word (34%):** The foreign word did not occur in the training corpus, so translation was not possible at all.
- **Unknown translation (14%):** The word occurred in the training corpus, but never alongside a translation that is needed in this context – or, the word alignment failed and failed to align the word to its correct translation, which often happens for rare words.

- **Tagging or parsing error (28%):** Errors in the tagger or parser that lead to incorrectly detected NP/PPs. These fragments could not properly translate into English, often because they need to be broken up and placed into distant parts of the sentence.
- **Complex syntactic restructuring (7%):** The translation of the NP/PP required a more involved syntactic restructuring that the model simply failed to accomplish.
- **Too long (6%):** A long NP/PP that has highly variable output. An acceptable translation could not be found in the 100-best list, but only further down the list.
- **Could not drop leading preposition (3%):** An acceptable translation of a prepositional phrase has to be a noun phrase (due to different subcategorization), but the phrase translation model is not able to drop the leading preposition.
- **Untranslatable (2%):** Some NP/PP are simply not translatable as NP/PP into English, as discussed in Section 1.6.
- **Other (6%)**

4.5.2 Reranking Failure

For 23% of all NP/PPs (313 out of 1362), an acceptable translation exists in the n-best list, but the maximum entropy re-ranker fails to pick it. Also for this portion of the test set, we carried out an error analysis. An overview of the results can be found in Table 4.8.

We looked at the translation that was picked by the system and tried to narrow down exactly what is wrong with it. It is hard to make such a qualitative assessment. It is to some part subjective to define what the problem is. We avoid more solution-oriented categories such us "insufficient context", or "not enough training data", but focus more on the symptoms.

The error categories we used come from a syntactic point of view. This does not mean that the cure has to be better syntactic modeling. A more empirically minded researcher might advise using "more data" or "a lower perplexity language model". The analysis does create a certain bias since it states what is wrong with the picked translation and suggest what has to be done to punish those choices, instead of stating what is right about the acceptable translations and what has to be done to encourage those.

For 44% of the NP/PPs with reranking failure, the error in the picked translation involves content words (nouns, adjectives, adverbs), while for 56% it involves only function words. 2% had other problems, such as reordering. The percentages do not add up to 100%, because multiple error categories may apply to a given NP/PP, but this is rarely the case.

The error categories in more detail:

Involving content words

- **Wrong word choice (16%):** This problem is related to word sense disambiguation. For a given word multiple valid translations exist (relating to different senses of the word), but in the given context the wrong translation is picked. This category only applies if the error could be corrected by changing one content word.

Error	**Frequency**
Involving content words	**44%**
Wrong word choice	16%
Content word mistranslated	4%
Wrong phrase choice	3%
Content dropped	13%
Content added	2%
Number of noun wrong	2%
Involving only function words	**56%**
Wrong phrase start	38%
Internal preposition choice	4%
Pronoun / anaphora	4%
Pronoun added or dropped	3%
Determiner added, dropped, wrong	2%
Function word phrase choice	2%
Function word mistranslated or dropped	2%
Preposition dropped	1%
Other	**2%**
Reordering wrong	1%
Other	1%

Table 4.8: Error analysis for NP/PPs for which the acceptable translation was not picked out of the n-best list (total 313)

- **Content word mistranslated (4%):** This is a worse case than wrong word choice, since the word translation that was chosen is so off-target that it never could be a good translation. The distinction to wrong choice is admitably somewhat fluid.
- **Wrong phrase choice (3%):** Some idiomatic phrases can have multiple translations (just as ambiguous words), and the wrong one was picked. For example: *etwas besseres* being translated as *a better standard*, but the correct translation being *something better*.
- **Content dropped (13%):** Some of the content in the input was dropped, resulting in a less informative or semantically wrong translation.
- **Content added (2%):** Some additional content was added, which changes the meaning.
- **Number of noun wrong (2%):** A plural noun was translated as singular, or vice versa. We addressed this problem with a syntactic feature (Section 4.3.2), but we could not completely rule out mistakes.

Involving only function words

- **Wrong phrase start (38%):** The initial sequence including preposition, determiner, and/or demonstrative pronoun was mistranslated, e.g., *in the city* instead of *to a city*. This is the largest error category. To get the phrase start right, often the sentence context is essential. Recall that in our experiment from Section 1.7 humans picked the wrong phrase start 9% of the time when given NP/PPs without context.
- **Internal preposition choice (4%):** A preposition in the middle of a NP/PP was translated wrong, e.g., *a report to the committee* instead of *a report by the committee*.
- **Pronoun / anaphora (4%):** A pronoun was translated wrong, since the foreign pronoun is ambiguous and it is unclear what it refers to (anaphora resolution). For instance, the German *ihr* could translate as *her*, *their*, or *your*.
- **Pronoun added or dropped (3%):** A pronoun was added or dropped. This happens surprisingly frequently for NP/PPs such as *we the people*. It probably due to the fact that these NP/PPs are often not translated literally in the original sentence-aligned corpus, leading to NP/PP alignments to, e.g., *the people*. The phrase translation system then learns that these pronouns can be dropped and added at will.
- **Determiner added, dropped, wrong (2%):** A determiner has to be included in an acceptable translation (or must not), but it is not (or is). This category also includes mistranslations such as *that* instead of *these*.
- **Function word phrase choice (2%):** Same as the phrase choice category that includes content words. Example: *weshalb* translated as *for this reason*, but correct translation is *why*.

- **Function word mistranslated or dropped (2%):** This includes mistranslations of function words that are not prepositions or pronouns. It includes some egregious examples such as the dropping of *no*.
- **Preposition dropped (1%):** A preposition in the middle of the sentence was dropped. We address this with a syntactic feature that preserves base PPs (Section 4.3.3), but occasional mistakes still occur.

Other

- **Reordering (1%):** It is rarely a problem that the NP/PP has to be restructured significantly and the system fails to find the right restructuring in the n-best. Recall from the previous section, that reordering problems are the cause of 7% of the cases, where no acceptable translation can be found in the n-best list. If such a reordering enters the n-best list, it will be picked out. Or, to phrase it more negatively, restructuring problems are so severe that we can hardly ever even produce a correct translation in the n-best for these NP/PPs.
- **Other (1%)**

To summarize, the main causes of error are wrong phrase starts (38%), mistranslated content words (20%) and dropped content (13%). Recall that in the human experiment from Section 1.7, roughly 90% of the error involved a wrong phrase start, and about 10% wrong word choice.

The error analysis leads to many ideas to improve NP/PP translation. It also gives some indication how valuable their contribution could be.

Chapter 5

Integration

In the previous two chapters, we described a subsystem for noun phrase translation. This chapter describes how such an independent component can be integrated into a general statistical machine translation system.

After discussing the design implications of such a separate subsystem (Section 5.1), we review the XML markup scheme developed by Germann [2003] for his greedy decoder (Section 5.2). This scheme lets a decoder use a pre-specified translation for a part of the input. We then describe how we equipped our beam search decoder with the same capability (Section 5.3) and extended the scheme to be able to pass a probability distribution over pre-specified translations to the decoder (Section 5.4).

5.1 Introduction

We separate the translation of NP/PP into a modular subsystem, while the translation of the rest of the sentence is handled by another translation system.

This is illustrated by Figure 5.1. The NP/PP *ein kleines Haus* is detected in the input and passed on to NP/PP translation subsystem. The subsystem translates the German NP/PP into the English NP/PP *a small house.* This translation is passed to the full sentence translation system, which is instructed to use this pre-specified translation. It translates the rest of the sentence, integrates the pre-specified translation and ends up with the translation *It is a small house.*

Dividing up the translation task into two parts that are addressed by different components has a number of design implications that a priori may hurt or benefit performance.

- The main argument for this design choice is that we can do a better job on translating NP/PPs, if we deal with it as a specialized problem. The performance of our dedicated NP/PP translation subsystem is better than the baseline statistical machine translation system.

- NP/PPs are forced to be translated as NP/PPs. This is mostly not a harmful restriction, as we argued earlier (Section 1.6): exceptions to this are rare and can be addressed by special handling. There is a benefit in being better able to control the output to conform to the correct syntactic structure. For instance, it is impossible in our set-up that the translation of a NP/PP may include a verb, which is not what we want anyway.

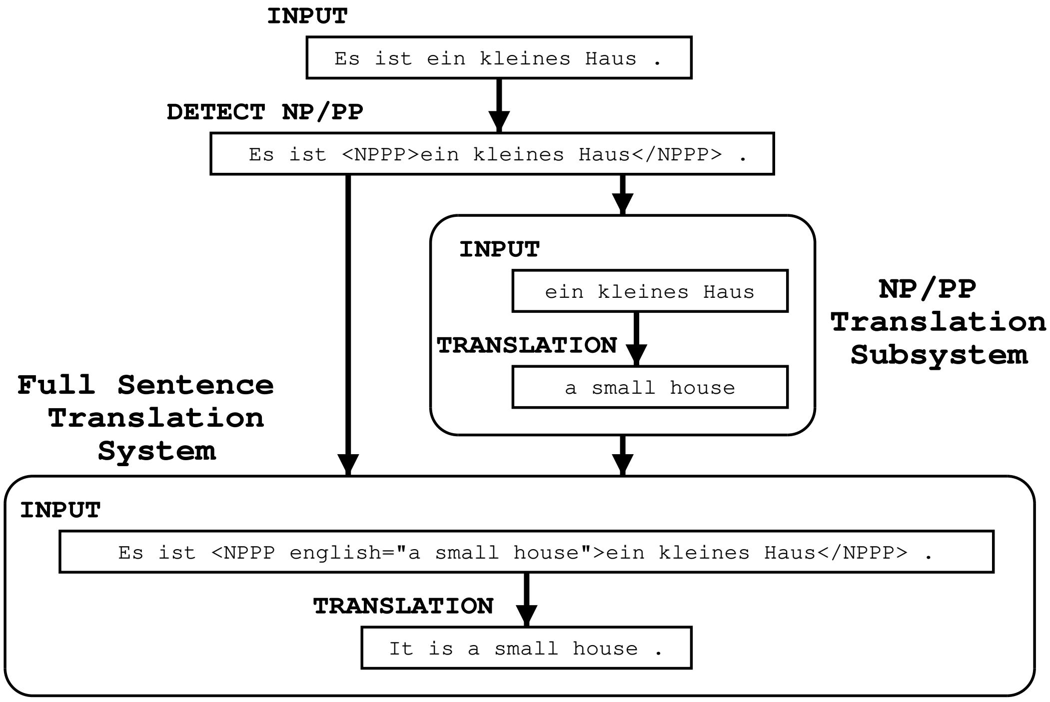


Figure 5.1: Integration of a NP/PP translation subsystem into a general full sentence machine translation system

- When separating out noun phrases, we lose their sentence context. This clearly is a concern that we would like to remedy. We may re-introduce the context in form of features. A powerful means to selecting between different translation choices is the language model. Since we use a language model for noun phrases in separation, we lose useful language model clues at the edges of the translated noun phrases. We will address this concern in Section 5.4.

- The NP/PP translation subsystem is trained on a corpus of noun phrase translations that we extracted from a parallel corpus. Recall from Section 3.2 that we could find corresponding noun phrases for 67.2% of German noun phrases with our corpus. This means a loss of 32.8% of the potential training data. Since the performance of statistical machine translation systems depends heavily on the amount of available training data, a loss of a third of the training data generally means a significant loss of performance. However, the reason that we could not align these noun phrases was that they do not cleanly align to noun phrases in English. Hence, excluding this noisy part of the training data may actually be beneficial.

- We have shown in our experiments for establishing a base statistical machine translation model that restricting phrasal translations to syntactic constituents is very harmful (see Section 3.3.4.4). By treating the syntactic constituent of type NP/PP in separation, we disallow phrasal translations that span over the boundaries of NP/PPs. This may mean the loss of valuable phrasal translation. As an example, a general phrase translation method may learn that *reist nach* translates to *travel to*. We disallow such phrasal translation, since the prepositions *to* are part of NP/PP, and the verbs are not. Note that this restriction is not as strong as the restriction in Section 3.3.4.4: Phrases (word sequences) within the NP/PPs and outside the NP/PPs do not have to be syntactic constituents.

- By detecting NP/PPs we make hard choices about the translation of the sentence. Errors during the detection process (such as parse errors) may cause problems down the road from which the system can not recover. The strength of the probabilistic approach is to not make such hard choices and allow for graceful decay when mistakes are made. See Section 5.5 on a remedy for this.

Ultimately, the judgment, whether the separation of NP/PP translation out of the translation task is beneficial or not, has to come from empirical evidence from experimentation. This is the focus of this chapter.

5.2 XML-Markup

While statistical machine translation methods cope well with many aspects of translation of languages, there are a few special problems for which better solution exist. One example is the translation of named entity, such as proper names, dates, quantities, and numbers.

5.2.1 Translating Named Entities

Consider the task of translating numbers, such as *17.55*. In order for a statistical machine translation system to be able to translate this number, it has to be observed in the training data. But even if it has been seen a few times, it is possible that the translation table learned for this "word" is very noisy.

Translating such numbers, especially among languages that share the same character set, is not a hard problem. It is therefore desirable to be able to tell the decoder up front how to translate such numbers.

The translation of named entities, for which other methods exist, was the motivation for the work by Germann [2003] to develop a XML markup scheme that, among other things, allows the specification of predefined translations in the input to a statistical machine translation system.

To continue our example of the number 17.55, this may take the following form:

```
Er erzielte <NUMBER english='17.55'>17,55</NUMBER> Punkte .
```

This modified input passes to the decoder not only the German words, but also that the third word, the number *17,55*, should be translated as *17.55*. The decoder accepts this translation and does not try to find a translation on its own.

The use of such predefined translations for named entities has been shown to improve the quality of a statistical machine translation system significantly, especially when only a small parallel corpus is available.

5.2.2 Translating Noun Phrases

We appropriate this method for our own purposes, the integration of noun phrase translation subsystem. As with the specification of translations for named entities, we specify translations for the noun phrases in the input, as they were generated by the NP/PP subsystem.

Consider the following example:

```
Es ist <NPPP english='a small house'>ein kleines Haus</NPPP> .
```

Here, the noun phrase *ein kleines Haus* was recognized in the input. It was passed to the NP/PP subsystem, which translated it as *a small house*. This translation is now specified in the modified input. This instructs the decoder to use this as a translation for this part of the sentence.

5.2.3 Experiments

In our first integration experiments, we integrate our NP/PP translation subsystem into an existing machine translation decoder: Rewrite[1], the freely available greedy decoder for the word-based translation model IBM Model 4. This decoder has the capability of accepting predefined translations in form of XML markup tags, as described above. It is the prototype implementation of the XML markup scheme.

[1] Available at http://www.isi.edu/licensed-sw/rewrite-decoder/

System	NP/PP Correct		BLEU
IBM Model 4	734	53.9%	0.172
Phrase Model	814	59.8%	0.187
Compound Splitting	853	62.6%	0.193
Re-Estimated Parameters	874	64.2%	0.195
Web Count Features	904	66.4%	0.197
Syntactic Features	914	67.1%	0.198

Table 5.1: Word-Based System: Increased accuracy of NP/PP translation leads to significantly better full sentence translation performance

So far, we evaluated the performance of the NP/PP translation subsystem by how many NP/PPs were translated correctly. The 1362 NP/PPs that we used in this evaluation are all the NP/PPs from 534 full sentences of length 5-15. To examine full sentence translation performance, we evaluate the translation performance of these sentences.

As evaluation metric, we use the BLEU score [Papineni et al., 2002], which is the current evaluation standard in statistical machine translation. It measures similarity to a reference translation. More precisely, it computes the geometric mean of unigram, bigram, trigram, and quadgram precision with a length penalty for too short translations.

We carried out experiments to compare the performance of a number of variations of the NP/PP subsystem. The results are displayed in Table 5.1. We already presented the accuracy judgments in Table 4.6 in Section 4.4.1. Note that the full sentence translation performance tracks the accuracy scores: When we add features to our system and NP/PPs translation accuracy increases from 53.2% to 65.5%, the BLEU score for full sentence translation also increases from 0.172 to 0.198.

Do these results represent an improvement over a baseline machine translation system that does not include special modeling for NP/PPs? To answer this question, we used the full sentence translation system to translate the whole sentence, without special handling of NP/PPs.

As Table 5.2 shows, we lose some performance by having a separate NP/PP translation subsystem: The baseline machine translation performance yields a BLEU score of 0.176 for our test set. When using exactly the same method for NP/PP translation, as for full sentence performance (IBM Model 4 with the greedy decoder), the score drops to 0.172. However, the benefits of additional modeling and features far out-weighs this drop: Recall from Table 5.1 that the most powerful NP/PP translation subsystem yields a BLEU score of 0.198 for full sentence translation performance.

To conclude: While there is a small performance loss when we separate out NP/PP translation into a subsystem a priori in the context of word-based statistical machine translation, our specialized NP/PP translation overcomes this and leads to much improved full sentence translation.

System	BLEU
No NP/PP subsystem	0.176
NP/PP subsystem with same modeling	0.172
NP/PP subsystem with all features	0.198

Table 5.2: Integrating the specialized NP/PP subsystem with all features leads to better translation performance than a baseline system without special NP/PP handling.

5.3 Phrase-Based Translation with NP/PP Subsystem

Germann [2003] developed an implementation of the XML markup scheme for a greedy decoder for a word-based statistical machine translation model. We implemented and extended it for a beam search decoder for a phrase-based statistical machine translation model.

5.3.1 Implementation

Marking a sequence of words and specifying a translation for them fits neatly into the framework of phrase-based translation. In a way, for a given phrase in the sentence, a translation is provided, which is in essence a phrase translation with translation probability 1. Only for the other parts of the sentence, translation options are generated from the phrase translation table.

Since the first step of the implementation of the beam search decoder is to collect all possible translation options, only this step has to be altered to be sensitive to specifications via XML markup. The core algorithm remains unchanged.

5.3.2 Experiments

We carried out the same experiments for a phrase-based translation model as done for the word-based translation IBM Model 4 in Section 5.2.3.

Again, the increase of performance of full sentence translation tracks the improvements in accuracy in NP/PP translation of the NP/PP subsystem. See Table 5.3 for details. Performance improves from 0.177 to 0.203.

Also, for phrase-based statistical machine translation, we want to examine, if the separation of NP/PP into a subsystem is harmful. As before, we train a system on all the data and perform no special treatment for noun phrases. In doing so, we obtain a score of 0.220 for our test set, much higher than the score that we get using our specialized NP/PP translation subsystem.

Table 5.4 compares the three major system designs: When separating out NP/PP translation into a subsystem and keeping the same modeling, we observe a dramatic drop in performance: From 0.220 to 0.193. Using all our features in our best NP/PP subsystem, we can raise the score significantly to 0.203, but not enough to overcome this drop.

System	NP/PP Correct		BLEU
IBM Model 4	724	53.9%	0.177
Phrase Model	800	59.8%	0.193
Compound Splitting	838	62.6%	0.199
Re-Estimated Parameters.	858	64.2%	0.202
Web Count Features	881	66.4%	0.204
Syntactic Features	892	67.1%	0.203

Table 5.3: Phrase-Based System: Increased accuracy of NP/PP translation leads to significantly better full sentence translation performance

System	BLEU
No NP/PP subsystem	0.220
NP/PP subsystem with same modeling	0.193
NP/PP subsystem with all features	0.203

Table 5.4: Separating NP/PP translation into a subsystem is more harmful for phrase-based translation

5.3.3 Discussion

We pointed out in the introduction to this chapter that there are disadvantages to separating out parts of the sentence for treatment by a separate subsystem, as opposed to keeping it all together.

The main problem is the loss of context. The language model is a powerful means to sort out correct lexical choices, but also reordering probabilities. In English, the verb is often directly followed by an object, which will be captured by a n-gram language model which helps to disambiguate word choices.

Of course, a trigram language model has some shortcomings. For instance, it does not capture relationships between more distant words. But abandoning it at the edges of noun phrases has serious consequences, as demonstrated by the results in Section 5.3.2.

5.4 Passing Probability Distribution of Translations

This section will describe how to address the disadvantages of loss of the support of the language model at the edges of NP/PPs. Instead of passing one best translation choice for the NP/PP subsystem to the full sentence translation system, we will now pass along a probability distribution over a set of possible translations.

System	BLEU
No NP/PP subsystem	0.220
NP/PP subsystem, no special modeling	0.193
NP/PP subsystem, passing top-1	0.203
NP/PP subsystem, passing top-100	0.215

Table 5.5: A probability distribution that includes up to 100 translations leads to better integration performance

5.4.1 Specification

We extend the XML markup scheme by allowing the specification of multiple English translation options along with translation probabilities.

To give an example:

```
Es ist <NPPP english='a small house|a little house' prob='0.6|0.4'> ein
kleines Haus</NPPP> .
```

Here, both *a small house* and *a little house* are passed along as possible translations, with the translation probabilities 0.6 and 0.4, respectively.

The scores that are passed along with the translations do not have to be probabilities in a strict sense, e.g., they do not have to add up to 1. They also do not include language model probabilities, but only the factor $p(f|e)$ for the marked part of the sentence.

5.4.2 Implementation

As in the case of passing a single best translation to the full sentence machine translation system, we treat the passed translations as translation options in the decoder. Now, they also include translation probability score, which corresponds to a phrase translation probability score.

The language model scores for all the words in the transfered translation and the phrase translations chosen by the decoder are factored in. This way, the language model helps with the selection of the best translation in the passed probability distribution, using the actual sentence context in which it is used.

5.4.3 Experiments

Table 5.5 displays the results of using this form of integration. We achieve much better results than when passing only the one best translation to the main system: Integrating the best subsystem this way yields an overall score of 0.215 instead of 0.203. Note that we gain this improvement solely from being able to take advantage of the language model when choosing NP/PP translation in the sentence context. However, the score is still below the score of 0.220 for the fully integrated system.

We still force translation of NP/PPs to NP/PPs, which may be almost always acceptable, but may not always conform to the stylistic choices in a text domain. Since we score against a reference translation from the domain, which makes the same stylistic choices, we get a lower score in these cases, even if our translation is acceptable as well.

5.5 Multi-Path Integration

For some NP/PPs, the use of the NP/PP translation subsystem is beneficial, for others the regular full sentence translation system would do a better job. This may depend on a number of factors, such as:

- **Translatability:** A NP/PP may not translate to a NP/PP in English.
- **Overlapping Phrasal Translation:** The full sentence translation system uses a sub phrase that overlaps the NP/PP and external context.
- **Good Features:** The features of the NP/PP translation subsystem score better translations higher.

Recognizing that the hard decision of breaking out certain words in the input sentence and relegating them to the NP/PP subsystem may be occasionally harmful, we now want to relax this decision. We allow the full sentence translation system to use the translations passed on by the NP/PP subsystem, but also to bypass this subsystem and use its own translations.

We call this multi-path integration, since we allow two pathways when translating noun phrases. For each NP/PP, the path may go through the results of the NP/PP translation subsystem, or we use translation options from the baseline full sentence translation system.

5.5.1 Implementation

Recall that the NP/PP translations that are provided to the full sentence translation subsystem are in the same format as the regular translation options from the phrase translation table. This made the integration of the NP/PP subsystem so straight-forward. Passing on an n-best of translations to the full sentence translation system is the same as looking up the phrase translation for a given phrase from the phrase translation table. In both cases, possible translations for a part of the input sentence with probabilities are provided to the decoder.

In other words: In multi-path translation, not only all the translation options from the full sentence phrase translation table can be used for translation any part of the sentence (including the NP/PPs), but also the translations generated by the NP/PP subsystem. It is left to the full sentence translation system to pick which translation option to use for each NP/PP.

One may conceive of many different weighting schemes to balance regular phrase translation table entries against NP/PP subsystem output. We only scale the probabilities of the NP/PP subsystem with an exponential parameter (which, as it turns out, has its best value close to one).

System	BLEU
No NP/PP subsystem	0.220
NP/PP subsystem, no special modeling	0.193
NP/PP subsystem, passing top-1	0.203
NP/PP subsystem, passing top-100	0.215
NP/PP subsystem, multi-path integration	0.223

Table 5.6: A probability distribution that includes up to 100 translations leads to better integration performance

5.5.2 Experiments

Table 5.6 includes the results for multi-path integration, along with the previous scores. Finally, we could improve over the baseline phrase-based translation system. However, the improvement of +0.003 is statistical significant only with a confidence of 80% (see Appendix A) for more discussion on statistical significance.

5.6 Conclusion

We showed that adding specialized modeling to NP/PP translation does not only increase translation accuracy of NP/PP translation itself, but it also helps to improve overall translation quality.

Separating out NP/PP translation into a subsystem carries a cost in terms of loss of context, which reduces translation quality. In the framework of word-based statistical machine translation, the improved quality of the NP/PP subsystem overcomes this cost and yields overall better translation quality (+0.022).

In the framework of phrase-based machine translation, this cost is much higher. We reduced this cost by passing a probability distribution of possible translations to the full sentence translation system instead of just the one best translation chosen by the subsystem and allowing multi-path back-off to the baseline system. So, even here we could show improvements over the baseline system, albeit to a lesser degree (+0.003) and only with limited statistical significance (80% confidence, as computed by pairwise bootstrap resampling, see Section A.3).

We created an environment for full sentence translation that recognizes and translates NP/PP as atomic units. This enables work on the sentence level that can take advantage of this: Translating a verb with its subcategorization behavior is much easier now that the subjects and objects can be detected and translated in isolation.

Chapter 6

Conclusions

We conclude this thesis with a summary of the main contributions of this work, a reflection on surprises and lessons learned, and review the shortcomings of our methods to point to possible future work. We made progress on noun phrase translation, but we have not achieved perfection. More interesting work is still awaiting the curious minds of future researchers.

6.1 Contributions

The main contribution of this thesis are:

- We defined noun phrase translation as a subtask of machine translation. We carried out empirical studies on how often NP/PPs can be translated in separation, the role of external context, and the sources of error of state-of-the-art methods.
- We presented the first empirical results on noun phrase translations. We described linguistic properties of NP/PP translation and exploited them with novel methods for compound splitting, use of web corpora, and syntactic features. We showed significant improvements in translation quality with these methods – both in terms of translation accuracy for noun phrase translation (from 53.9% to 67.1%), as well as full sentence translation (see Table 6.1).
- We introduced a novel framework of dealing with subtasks like noun phrase translation, consisting of detection of NP/PPs, collection of a training corpus, translation by maximum entropy reranking, and integration via XML into a full sentence translation system.

System	Word-Based MT	Phrase-Based MT
Baseline	0.176	0.220
Using NP/PP subsystem	0.198	0.223

Table 6.1: Full sentence translation improves using our NP/PP translation subsystem, by +0.022 for word-based MT, and by +0.003 for phrase-based MT.

- We contributed to the understanding of phrase-based translation by defining a generalized framework that includes previously proposed methods and carried out an extensive comparative investigation of phrase translation methods.

6.2 Surprises

When researching the properties of a problem, and inventing solutions, one often encounters surprising results. While the most common surprise – that a well-conceived method grounded in intuition does not perform as well as expected – is the cause for great frustration, these surprises do all contribute something very valuable and ultimately justify the endeavor of research: They create new knowledge.

In this section, we want to highlight a few findings that were especially surprising to us, and should therefore be especially valuable to the reader.

6.2.1 NP/PP Translation as Subtask

It was not obvious from the outset that NP/PP translation can be separated out into a subtask of statistical machine translation. The study described in the introduction (Section 1.6) gives some evidence, but it is ultimately confirmed by the results on noun phrase translation proves that there are only few exceptions where the translation of a NP/PP as a unit is not possible (Section 4.5).

One might object that German and English are very related languages with similar syntactic structure, which is in fact a quite questionable statement due to the strict-order nature of English vs. the free-order, rich-morphology nature of German. But we believe that since NP/PP are the grammatical category in language used for naming objects and concepts in the world, their translation should be similar atomic for other language pairs. Preliminary studies on other language pairs (Chinese-English, Portuguese-English) gave some evidence to confirm this intuition.

6.2.2 Compound Splitting

The splitting of compounds in German is a necessary step for any machine translation system that translates from this language. Therefore, we spent a significant amount of effort to come up with data-driven methods for this problem (Section 4.1).

Since the problem turned out mostly one of the granularity of the split, and less of widely different splitting alternatives, a fairly simple frequency-based method turned out to be sufficient in the context of a phrase-based machine translation system.

There are related problems of granularity in machine translation, such as word segmentation in Chinese. Text in Chinese does not have spaces between words, so either word segmentation tools have to be used, or the text is broken up into characters. Splitting up Chinese text into characters is analogous to the eager compound splitting method. Using this simple solution to Chinese word segmentation does not dramatically decrease performance. This also demonstrates how robust phrase-based translation statistical machine translation is to such issues of granularity.

6.2.3 Syntax and Phrase-Based Machine Translation

One finding of great concern for the integration of syntax and phrase-based machine translation came up in our work on the base model. The "phrases" used in the phrase-based model are simply consecutive sequences of words without any linguistic motivation.

We suspected that syntactic phrases (such as base noun phrases) would be usually more reliable, but our experiments showed that this is not the case. Relying only on syntactic phrases led to devastating results, and weighting syntactic phrases stronger was also not beneficial (Section 3.3.4.2, 3.3.4.4).

This finding is troublesome for translation models based on syntax tree reordering. The results suggest that the mapping of constituent trees to constituent trees is not as powerful as the mapping of any string sequences. While such syntax-based machine translation models do have other benefits, especially the explanation of transformations that are syntactic in nature, they have to overcome this disadvantage, which may be a too steep hill to climb.

6.2.4 Integration

While the integration of our NP/PP translation subsystem into word-based machine translation proved to be of great benefit, the integration into a phrase-based system was more difficult. Phrase-based machine translation is already a pretty good baseline. As we just mentioned, cutting possible phrase translations along syntactic lines is generally harmful.

As a consequence, we did incur a great cost for this separation of NP/PP translation from full sentence translation (a cost of -0.027, according to Table 5.4 on page 78). We overcame this cost by passing an n-best list and allowing multi-path integration, but the overall improvement was less impressive than the one for the word-based translation system.

6.3 Shortcomings and Future Work

In this section, we review shortcomings of our approach, as it is implemented at this point, and re-examine the main sources of errors. These present challenges for future work. We will sketch out a few ideas for smaller and larger projects to address these challenges.

6.3.1 Acquiring Translation Knowledge for Unknown Words

Unknown words and translation are the main type of error for NP/PPs without an acceptable translation in the n-best list (recall the error analysis in Table 4.7 on page 67). Overall, this causes the NP/PP translation subsystem to fail on 5% of the NP/PPs.

The nice thing about unknown words is that the problem is easy to detect. If we do not know how to translate a word because we have never seen it before, we immediately know it. Detection can also be extended to rare words that occur only a few times in training and where we thus have little confidence in its translation table.

There has been some work in the area of learning translation lexicons from comparable corpora, including or own [Koehn and Knight, 2000, 2001, 2002b]. If we know in which foreign contexts a foreign word is used, we can look for similar English contexts and detect possible English translations.

There has also been work on fishing for parallel corpora on the web that include the word in question [Nagata et al., 2001]. Related work is the search for unknown phrase translations on the web [Cao and Li, 2002] or in large corpora [Munteanu and Marcu, 2002].

None of this work has been integrated into a machine translation system and evaluated for improvements of translation performance.

6.3.2 Richer Model for Phrase Starts

During reranking, the main cause of error is picking the wrong phrase start, accounting for almost 10% of the overall error (see Table 4.8 on page 69).

The obvious answer to the inability of the isolated NP/PP translation subsystem to pick the proper phrase start is to provide the sentence context to the subsystem. During integration, we will address this in form of the language model in Section 5.4. However, a trigram language model is not sufficient for dependencies between words that are distant from each other.

Typically, the phrase start depends on the foreign phrase start, the content of the foreign NP/PP, the foreign attachment of the NP/PP (in most cases, the subcategorization of the verb), the content of the English NP/PP and the English attachment of the NP/PP. Integrating all the attachment dependencies into a phrase start translation model seems to be an appropriate strategy to address this.

6.3.3 Word Choice

Roughly 5% of the overall error is due to erroneous word choice, i.e., picking of the wrong translation for an ambiguous content word.

There is vast research on word sense disambiguation, a problem related to choosing the right word translation. This work has stimulated only little research in machine translation, and none of the results have been too promising [Berger et al., 1996; Varea et al., 2001]. This may be due to the fact that the language model already is a powerful means for picking the proper word sense and the main problem is insufficient training data (see our earlier work [Koehn and Knight, 2001]).

But it may also be most work in this area tried to integrate word choice into the decoder and is therefore limited to local context, which is already addressed by language model and phrase translation. In a reranking framework, we can take advantage of a much bigger window. Researchers in word sense disambiguation work typically with windows as large as 50 words around the target word.

6.3.4 Content Preservation

The third large source of error during reranking is dropped and added content (roughly 4% of overall error). Nouns, adjectives and adverbs should generally not be dropped during translation and we can imagine a syntactic feature to address this problem. But it is not as simple as requiring that every noun should be translated as a noun, since there are many exceptions to this, e.g., German compounds typically translate into multiple English nouns.

6.3.5 Integration

The integration of the noun phrase subsystem into the full sentence translation system can still be improved. We expect improvements from using sentence context within the NP/PP translation subsystem, as suggested above.

One main problem is that we cut out the NP/PPs from the sentence, thereby eliminating possible phrase translations that span the boundaries of the NP/PPs. This may be alleviated by using such overlapping phrases as support for translation choices outside and inside the NP/PPs. Work by Stephan Vogel on using such overlapping phrases has shown improvements in general[1], and may be of especially beneficial use for the integration of NP/PP translation.

6.3.6 Clause Structure

Our work in NP/PP translation puts us in a much better position to address the next level of syntactic structure: clauses (recall Figure 1.1 on page 2). The translation of clauses may cause reordering and the insertion and deletion of function words.

While English sentence are ordered subject-verb-object (SVO), other languages may have different ordering. Most sentences in German, a relatively free word order language, are SVO or OVS.

To deal with this during translation into English, it would be useful to have a transformation rule such as:

- `object - verb - subject` → `subject - verb - object`

The noun phrases have to be tagged for their syntactic role, which is indicated by case or prepositions. Besides general reordering rules such as the one above, reordering may also depend on the different subcategorization of a foreign verb and its translation.

In German, the verb may be placed at different positions in the sentence. A number of possible verb placements are:

- `subject - verb - objects - verb-particle`
- `subject - aux - objects - verb`
- `adverb - verb - subject - object`
- `subject - objects - verb` (relative clause)

[1] Personal communications with Stephan Vogel

Translation of these structures may require the movement of the verb into a different position and may even require the collection of the main verb complex (auxiliary + verb) from different parts of the sentence and its translation into a single verb.

Languages differ in their use of pronouns. Pro-drop languages may have clause structures without a subject, when the English equivalent requires a pronoun in the subject position.

- `verb - object` → `pronoun - verb - object`
  Hablo ingles → *I speak English*

A translation of a verb may have additional subcategorization slots, which have to be filled with pronouns, e.g.:

- `pronoun - verb - pronoun - object` → `pronoun - verb - object`
  Es macht mir Spass → *I have fun*

- `pronoun - verb - object` → `subject - verb`
  Es scheint die Sonne → *The sun is shining*

Clause structure does not only explain ordering, dropping and adding of arguments. The realization of translated arguments (subject, objects) may also depend on the verb: as noun phrases or prepositional phrases with certain prepositions. Recall the examples from Section 1.7, where the realization of the object depends on the verb:

- *Ich ziele <u>auf den Mann</u>* → *I aim <u>at the man</u>*

- *Ich gehe <u>auf den Mann</u> zu* → *I walk <u>to the man</u>*

- *Ich gehe <u>auf den Mann</u> zu* → *I approach <u>the man</u>*

This short description of issues in clause structure makes clear, how dedicated modeling may help to improve translation quality in the same way it dedicated modeling of noun phrases helped translation quality. Our NP/PP translation subsystem fits nicely as a building block into such modeling. We expect to see interesting work in this area in the future.

Bibliography

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., and Yarowsky, D. (1999). Statistical machine translation. Technical report, John Hopkins University Summer Workshop `http://www.clsp.jhu.edu/ws99/projects/mt/`.

Al-Onaizan, Y. and Knight, K. (2002). Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Alshawi, H., Bangalore, S., and Douglas, S. (2000). Learning dependency translation models as collection of finite-state head transducers. *Computational Linguistics*, 26(1):45–60.

Arnold, D. J., Balkan, L., Meijer, S., Humphreys, R. L., and Sadler, L. (1994). *Machine Translation: an Introductory Guide*. Blackwells–NCC.

Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–69.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories Sozopol.*

Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP)*.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4).

Brown, P., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Rossin, P. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):76–85.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

Brown, R. D. (2002). Corpus-driven splitting of compound words. In *Proceedings of the Ninth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

Cao, Y. and Li, H. (2002). Base noun phrase translation using web data and the EM algorithm. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Chen, S. F. and Goodman, J. (1998). An emprirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

Collins, M. (1997). Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics (ACL).*

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap.* CRC Press.

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL).*

Eppstein, D. (1994). Finding the k shortest paths. In *Proc. 35th Symp. Foundations of Computer Science*, pages 154–165. IEEE.

Finkler, W. and Neumann, G. (1998). Morphix. A fast realization of a classification-based approach to morphology. In *4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop - Wissensbasierte Sprachverarbeitung.*

Germann, U. (2003). Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computation Linguistics (HLT-NAACL).*

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL).*

Gildea, D. (2003). Loosly tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL).*

Hedlund, T., Keskustalo, H., Pirkola, A., Airio, E., and Järvelin, K. (2001). Utaclir @ CLEF 2001 - effects of compound splitting and n-gram techniques. In *Second Workshop of the Cross-Language Evaluation Forum (CLEF), Revised Papers.*

Hutchins, W. J. and Somers, H. L. (1992). *An Introduction to Machine Translation.* Academic Press, London.

III, E. H. N. and Mitamura, T. (1992). The kant system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Imamura, K. (2002). Application of translation knowledge acquired by hierarchical phrase alignment for pattern-based mt. In *Proceedings of the Ninth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI).*

Jelinek, F. (1998). *Statistical Methods for Speech Recognition*. The MIT Press.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Koehn, P. (2002). Europarl: A multilingual corpus for evaluation of machine translation. Unpublished, `http://www.isi.edu/~koehn/europarl/`.

Koehn, P. and Knight, K. (2000). Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proceedings of Annual Meeting of the American Association of Artificial Intelligence (AAAI)*.

Koehn, P. and Knight, K. (2001). Knowledge sources for word-level translation models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Koehn, P. and Knight, K. (2002a). ChunkMT: Machine translation with richer linguistic knowledge. Unpublished.

Koehn, P. and Knight, K. (2002b). Learning a translation lexicon from monolingual corpora. In *40th Annual Meeting of the Association of Computational Linguistics (ACL), Workshop of Unsupervised Lexicon Construction*.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Langer, S. (1998). Zur Morphologie und Semantik von Nominalkomposita. In *Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*.

Langkilde, I. (2000). Forest-based statistical sentence generation. In *Proceedings of Annual Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL)*.

Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In *6th International Conference on Spoken Language Processing (ICSLP)*.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Melamed, I. D. (2003). Multitext grammars and synchronous parsers. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Mitchell, T. M. (1997). *Machine Learning*. WCB/McGraw-Hill.

Monz, C. and de Rijke, M. (2001). Shallow morphological analysis in monolingual information retrieval for Dutch, German, and Italian. In *Second Workshop of the Cross-Language Evaluation Forum (CLEF), Revised Papers*.

Munteanu, D. S. and Marcu, D. (2002). Processing comparable corpora with bilingual suffix trees. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nagata, M., Saito, T., and Suzuki, K. (2001). Using the web as a bilingual dictionary. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. (1992). *Machine Translation: A knowledge-based approach*. Morgan Kaufman, San Mateo.

Och, F. J. (1998). Ein beispielsbasierter und statistischer Ansatz zum maschinellen Lernen von natürlichsprachlicher übersetzung. Master's thesis, Universität Erlangen-Nürnberg.

Och, F. J. (1999). An efficient method for determining bilingual word classes. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 71–76.

Och, F. J. (2002). *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, RWTH Aachen, Germany.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 20–28.

Och, F. J., Ueffing, N., and Ney, H. (2001). An efficient A* search algorithm for statistical machine translation. In *Data-Driven Machine Translation Workshop*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176(W0109-022), IBM Research Report.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Russel, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey.

Schafer, C. and Yarowski, D. (2003). Statistical machine translation using coercive two-level syntactic transduction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Schmidt, H. and Schulte im Walde, S. (2000). Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Seymore, K. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of Eurospeech*.

Somers, H. (1999). Review article: Example-based machine translation. *Machine Translation*, 14:113–157.

Tillmann, C. (2001). *Word Re-Ordering and Dynamic Programming based Search Algorithm for Statistical Machine Translation*. PhD thesis, RWTH Aachen, Germany.

Tillmann, C. (2003). A projection extension algorithm for statistical machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ueffing, N., Och, F. J., and Ney, H. (2002). Generation of word graphs in statistical machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Varea, I. G., Och, F. J., and Casacuberta, F. (2001). Refined lexicon models for statistical machine translation using a maximum entropy approach. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Venugopal, A., Vogel, S., and Waibel, A. (2003). Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3).

Yamada, K. (2002). *A syntax-based translation model.* PhD thesis, Department of Computer Science, University of Southern California, Los Angeles.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL).*

Appendix A

Statistical Significance

We evaluated the performance of our system on a sample test set, which is randomly drawn from held-out data that is not used in training. The size of the test depends on a number of factors, such as the amount of available data and the computational cost to process it repeatedly for different parameter settings.

Given the desire to have reliable results, we want a test set as large as possible. The bigger the test set, the more likely results on it will represent results on any data of this kind.

A.1 Confidence Intervals

If a measured test score (such as translation accuracy of a set of NP/PPs) represents the true score of a system is a common question in empirical studies. According to statistical theory [Mitchell, 1997], if the test samples are drawn randomly, the true error lies with 95% probability in the interval

$$e \pm 1.96\sqrt{\frac{e(1-e)}{n}} \tag{A.1}$$

where e is the measured error and n the number of samples.

In the results reported in Section 4.4.1, we used a test set of 1362 NP/PPs, so n=1362. In Table A.1, we display not only the accuracy scores, but also the confidence values according to the preceding formula.

These intervals are commonly used to check, if one method achieves superior results to another method. In our case, our fully employed system has (with 95% probability) a true error in the interval 65.8-68.4%, while the baseline system has a true error of 52.5-55.3%. These numbers show that we has significantly improved performance.

A.2 Bootstrap Resampling

We measure full sentence translation performance using the BLEU score. Unfortunately, since the score is derived in a rather convoluted way, we cannot easily apply the formula above. Whenever an analytical estimate of the confidence interval cannot be derived,

System	Accuracy	Interval
IBM Model 4	53.9%	52.5-55.3%
Phrase Model	59.8%	58.5-61.1%
Compound Splitting	62.6%	61.3-63.9%
Re-Estimated Parameter	64.2%	62.9-65.5%
Web Count Features	66.4%	65.1-67.7%
Syntactic Features	67.1%	65.8-68.4%

Table A.1: Confidence intervals for NP/PP translation accuracy (see Section 4.4) as computed by Formula A.1

System	BLEU	Interval
IBM Model 4	0.172	0.159-0.186
Phrase Model	0.187	0.173-0.200
Compound Splitting	0.193	0.179-0.208
Re-Estimated Parameter	0.195	0.181-0.210
Web Count Features	0.197	0.182-0.212
Syntactic Features	0.198	0.183-0.212

Table A.2: Confidence intervals for NP/PP translation accuracy (word-based machine translation, see Section 5.2.3) as computed by bootstrap resampling

one may alternately use the method of bootstrap resampling. A good introduction to bootstrap resampling is provided by Efron and Tibshirani [1994].

To empirically estimate the variance in a test sample, we would repeatedly randomly draw a sample and check its score. If we, say, draw 1000 test samples this way, we can sort the resulting scores and estimate the 95% confidence interval, by dropping the 25 lowest and 25 highest scores. The interval that the remaining 950 sample scores span corresponds to the 95% confidence interval computed analytically in the previous section.

Unfortunately, we often cannot collect 1000 test samples this way. If we could, we would just use a 1000 times bigger test sample to begin with and would have much more reliable results. Therefore, we draw the 1000 test samples randomly from the initial test sample. That means in our case, for each of the 1000 test samples, we draw 1362 NP/PPs from the test set of 1362 NP/PPs with replacement.

Table A.2 shows the confidence intervals for the machine translation performance on the full sentence task, obtained by using bootstrap resampling in this way.

On machine translation performance, all the confidence intervals overlap. The confidence intervals we computed so far give us some indicates how well the score of a system would look like on any sample of test data of the same type. However, what we are really interested in is if one system performs better than another system on any test set.

System	BLEU	Better than					
		(1)	(2)	(3)	(4)	(5)	(6)
(1) IBM Model 4	0.172	–	0%	0%	0%	0%	0%
(2) Phrase Model	0.187	100%	–	0%	0%	0%	0%
(3) Compound Splitting	0.193	100%	100%	–	7%	2%	2%
(4) Re-Estimated Parameter	0.195	100%	100%	93%	–	13%	10%
(5) Web Count Features	0.197	100%	100%	98%	87%	–	35%
(6) Syntactic Features	0.198	100%	100%	98%	90%	65%	–

Table A.3: Confidences that systems do better than others, as computed by pairwise bootstrap resampling

A.3 Pairwise Bootstrap Resampling

Comparing the confidence intervals above leads to a much too conservative error estimate: When comparing two systems, we use the same test set. One can easily see that if we would compare two system on different test sets, the performance varies much more, just because one system may run on a simpler test set than the other.

Therefore, we propose to apply pairwise bootstrap resampling to address the fact that we test the systems on the same test set. Again, we collect 1000 test sample sets as above. We then compute the score for two systems. If for at least 950 test samples, one system does better than the other, this gives empirical evidence that it is better with 95% confidence.

Table A.3 shows for each system with how much confidence it is better than the baseline system IBM Model 4 and the system in the previous line.

The numbers show that our NP/PP translation subsystems lead to statistically significantly improved machine translation performance when integrated with word-based machine translation. This cannot be shown for every set of added features, e.g., we can say only with 65% certainty that the added syntactic features help on top of all the other features, when integrated into a full sentence translation system.

We computed statistical significance in the same way to evaluate the improvement of full sentence translation by adding the NP/PP translation subsystem to the phrase-based translation system. Here, the confidence is 80%.

Appendix B

Additional Properties of the Base Model

In this section, we present a number of additional results for the base model we used in our work – the phrase based machine translation system (see Section 3.3). Some of these findings are published in the paper "Statistical Phrase Based Translation" [Koehn et al., 2003].

As in the other experiments in this thesis, we used the freely available Europarl corpus[1] to carry out experiments. Recall that this corpus contains over 20 million words in each of the eleven official languages of the European Union, covering the proceedings of the European Parliament 1996-2001. 1755 sentences of length 5-15 were reserved for testing.

In all experiments in Sections 3.3.4.5-B.5 we translate from German to English. We measure performance using the BLEU score [Papineni et al., 2001], which estimates the accuracy of translation output with respect to a reference translation.

B.1 Maximum Phrase Length

How long do phrases have to be to achieve high performance? Figure B.1 displays results from experiments with different maximum phrase lengths. Word alignment induced phrases (WAIPh) are used. Surprisingly, limiting the length to a maximum of only three words per phrase already achieves excellent performance. Learning longer phrases does not yield much improvement, and occasionally leads to worse results. Reducing the limit to only two, however, is clearly detrimental.

Allowing for longer phrases increases the phrase translation table size (see Table B.1). The increase is almost linear with the maximum length limit. Still, none of these model sizes cause memory problems.

B.2 Lexical Weighting

The phrase translation probability distribution is estimated by maximum likelihood estimation. Since this is not very reliable for low counts, and no smoothing is performed, we may want to look for additional ways to validate the quality of a phrase translation pair. One way is to check how well its words translate to each other. For this, we need a

[1] The Europarl corpus is available at `http://www.isi.edu/~koehn/europarl/`

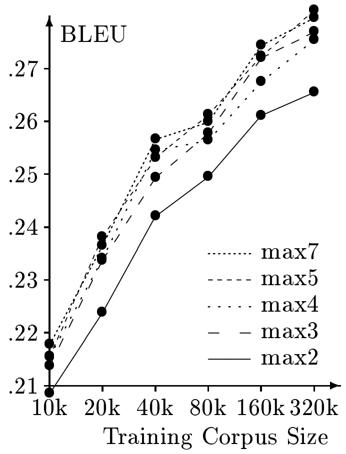


Figure B.1: Different limits for maximum phrase length show that length 3 is enough

Max. Length	Training corpus size					
	10k	20k	40k	80k	160k	320k
2	37k	70k	135k	250k	474k	882k
3	63k	128k	261k	509k	1028k	1996k
4	84k	176k	370k	736k	1536k	3152k
5	101k	215k	459k	925k	1968k	4119k
7	130k	278k	605k	1217k	2657k	5663k

Table B.1: Size of the phrase translation table with varying maximum phrase length

lexical translation probability distribution $w(f|e)$. We estimated it by relative frequency from the same word alignments as the phrase model.

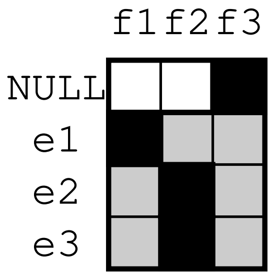
$$w(f|e) = \frac{\text{count}(f,e)}{\sum_{f'} \text{count}(f',e)}$$

A special English NULL token is added to each English sentence and aligned to each unaligned foreign word.

Given a phrase pair $\bar{f}, \bar{e}$ and a word alignment a between the foreign word positions $i = 1, ..., n$ and the English word positions $j = 0, 1, ..., m$, we compute the lexical weight p_w by

$$p_w(\bar{f}|\bar{e}, a) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(f_i|e_j)$$

See Figure B.2 for an example.



$$
\begin{aligned}
p_w(\bar{f}|\bar{e}, a) &= p_w(f_1 f_2 f_3|\bar{e}, a) \\
&= p_w(f_1|\bar{e}, a) \times p_w(f_2|\bar{e}, a) \times p_w(f_3|\bar{e}, a) \\
&= w(f_1|e_1) \times (w(f_2|e_2) + w(f_2|e_3))/2 \times w(f_3|\text{NULL})
\end{aligned}
$$

Figure B.2: Lexical weight p_w of a phrase pair $(\bar{f}, \bar{e})$ given an alignment a and a lexical translation probability distribution $w(\cdot)$

If there are multiple alignments a for a phrase pair $(\bar{f}, \bar{e})$, we use the one with the highest lexical weight:

$$p_w(\bar{f}|\bar{e}) = \max_a p_w(\bar{f}|\bar{e}, a)$$

We use the lexical weight p_w during translation as a additional factor. This means that the model $p(f|e)$ is extended to

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1}) p_w(\bar{f}_i|\bar{e}_i)^\lambda$$

The parameter λ defines the strength of the lexical weight p_w. Good values for this parameter are around 0.25.

Figure B.3 shows the impact of lexical weighting on machine translation performance. In our experiments, we achieved improvements of up to 0.01 on the BLEU score scale. Again, all phrases consistent with the word alignment are used (Section 3.3.4.1).

Note that phrase translation with a lexical weight is the special case of the alignment template model [Och et al., 1999] with one word class for each word. Our simplification has the advantage that the lexical weights can be factored into the phrase translation table beforehand, speeding up decoding.

B.3 Segmentation and Word Cost

An implicit first step in the phrase translation process is the segmentation of the input word sequence into phrases. So far, we ignored this step in our generative story, thus assuming a uniform probability distribution over all possible segmentations.

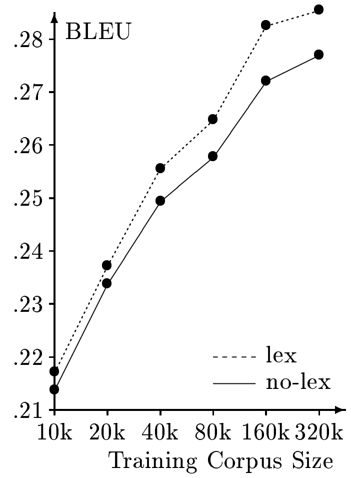


Figure B.3: Lexical weighting (lex) improves performance.

There are many ways to model the segmentation of the input. One apparent flaw of treating all segmentations equally is that in finer grained segmentation, more translation probabilities are factored in. In addition, translation probability distribution for smaller phrases are flatter, leading to larger costs.

Hence, we might want to introduce a bias for finer (or coarser) granularity of the segmentation. This can be done with a cost factor for each phrase translation. If this factor is larger than one, we bias toward more, smaller phrases. Coarser segmentation is preferred with a factor smaller than one.

The experimental results seem to suggest that a bias toward smaller phrases is beneficial. However, the main effect of using smaller phrases is generating more words in the output. Since the unbiased system produces too short output, this is advantageous.

We can also directly introduce a bias toward shorter or longer output by giving a penalty or benefit for each word produced. While this is less principled in terms of the generative story of the machine translation process, it is a fairly easy and effective means.

Figure B.4 shows the effect of both the phrase cost factor (for each phrase translation) and the word cost factor (for each English word generated). The graph plots the length ratio of system output and reference output against the resulting BLEU score. Peak performance can be obtained, if this ratio is close to 1. Both phrase cost and word cost are parameters which allow us to easily tune the verbosity of the system.

B.4 Phrase Extraction Heuristic

Recall from Section 3.3.4.1 that we learn phrase pairs from word alignments generated by Giza++. The IBM Models that this toolkit implements only allow at most one English word to be aligned with a foreign word. We remedy this problem with a heuristic approach.

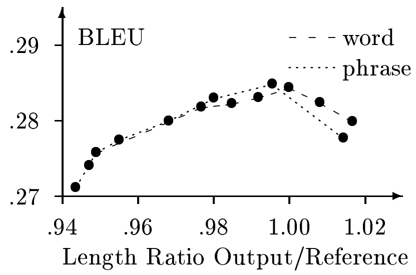


Figure B.4: A cost factor for each generated word (word) or for each phrase translation (phrase) can be used to calibrate the output length.

First, we align a parallel corpus bidirectionally – foreign to English and English to foreign. This gives us two word alignments that we try to reconcile. If we intersect the two alignments, we get a high-precision alignment of high-confidence alignment points. If we take the union of the two alignments, we get a high-recall alignment with additional alignment points.

We explore the space between intersection and union with expansion heuristics that start with the intersection and add additional alignment points. The decision which points to add may depend on a number of criteria:

- In which alignment does the potential alignment point exist? Foreign-English or English-foreign?
- Does the potential point neighbor already established points?
- Does "neighboring" mean directly adjacent (block-distance), or also diagonally adjacent?
- Is the English or the foreign word that the potential point connects unaligned so far? Are both unaligned?
- What is the lexical probability for the potential point?

The base heuristic [Och et al., 1999] proceeds as follows: We start with intersection of the two word alignments. We only add new alignment points that exist in the union of two word alignments. We also always require that a new alignment point connects at least one previously unaligned word.

First, we expand to only directly adjacent alignment points. We check for potential points starting from the top right corner of the alignment matrix, checking for alignment points for the first English word, then continue with alignment points for the second English word, and so on. This is done iteratively until no alignment point can be added anymore. In a final step, we add non-adjacent alignment points, with otherwise the same requirements.

Figure B.5 shows the performance of this heuristic (base) compared against the two mono-directional alignments (e2f, f2e) and their union (union). The figure also contains

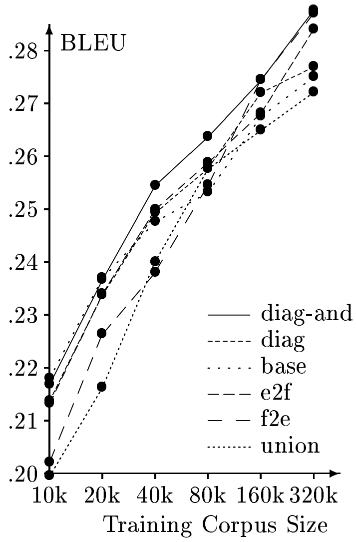


Figure B.5: Different heuristics to symmetrize word alignments from bidirectional Giza++ alignments

two modifications of the base heuristic: in the first (diag) we also permit diagonal neighborhood in the iterative expansion stage. In a variation of this (diag-and), we require in the final step that both words are unaligned.

The ranking of these different methods varies for different training corpus sizes. For instance, the alignment f2e starts out second to worst for the 10,000 sentence pair corpus, but ultimately is competitive with the best method at 320,000 sentence pairs. The base heuristic is initially the best, but then drops off.

The discrepancy between the best and the worst method is quite large, about 0.02 BLEU. For almost all training corpus sizes, the heuristic diag-and performs best, albeit not always significantly.

Recently, alternative heuristics for phrase extraction have been proposed by Tillmann [2003] and Venugopal et al. [2003]. This problem is still an active research topic, where we expect advances in the future.

B.5 Simpler Underlying Word-Based Models

The initial word alignment for collecting phrase pairs is generated by symmetrizing IBM Model 4 alignments. Model 4 is computationally expensive, and only approximate solutions exist to estimate its parameters. The IBM Models 1-3 are faster and easier to implement. For IBM Model 1 and 2 word alignments can be computed efficiently without relying on approximations. For more information on these models, please refer to Brown et al. [1993]. Again, we use the heuristics from the Section B.4 to reconcile the monodirectional alignments obtained through training parameters using models of increasing complexity.

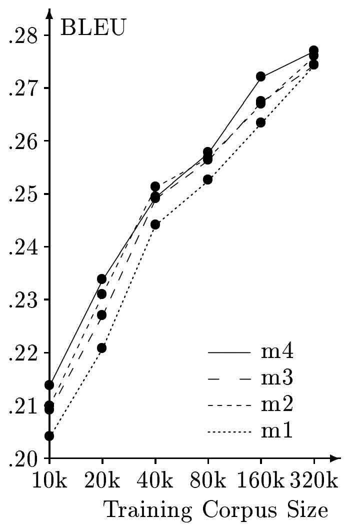


Figure B.6: Using simpler IBM models for word alignment does not reduce performance much

How much is performance affected if we base word alignments on these simpler methods? As Figure B.6 indicates, not much. While Model 1 clearly results in worse performance, the difference is less striking for Model 2 and 3. Using different expansion heuristics during symmetrizing the word alignments has a bigger effect.

We can conclude from this that high quality phrase alignments can be learned with fairly simple means. The simpler and faster Model 2 provides similar performance to the complex Model 4.

B.6 Impact of Language Model

In all our experiments we used the same training data for learning the translation model and the language model. While this is more appropriate to the reality of having training corpora of different sizes, it raises the question that improvement in translation quality may only have improved due to a better language model.

We checked this concern by keeping a fixed language model trained on the 320,000 word translation corpus, while varying the corpus size for learning the translation model. Figure B.7 suggests that while the language model has an impact, performance increases are mostly due to the improved translation model.

B.7 Conclusions

We created a framework (translation model and decoder) that enables us to evaluate and compare various phrase translation methods. Our results show that phrase translation gives better performance than traditional word-based methods. We obtain the best results

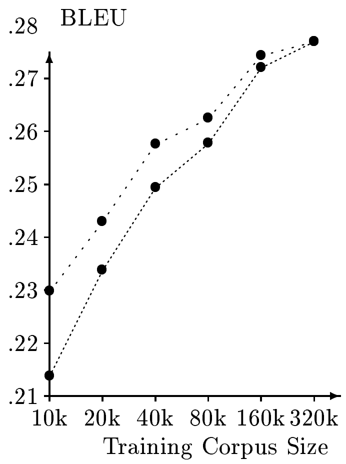


Figure B.7: Having just a larger language model helps

Language Pair	Model4	Phrase	Lex
English-German	0.204	0.236	0.245
French-English	0.279	0.329	0.339
English-French	0.256	0.315	0.324
Finnish-English	0.218	0.274	0.281
Swedish-English	0.314	0.346	0.355
Chinese-English	0.119	0.140	0.142

Table B.2: Confirmation of our findings for additional language pairs (measured with BLEU): Phrase-Based MT performs better than the word-based IBM Model 4 and lexicalization helps.

even with small phrases of up to three words. Lexical weighting of phrase translation helps.

We validated our findings for additional language pairs. Table B.2 displays some of the results. For all language pairs the phrase model (based on word alignments, Section 3.3.4.1) outperforms IBM Model 4. Lexicalization (Lex) always helps as well.

Straight-forward syntactic models that map constituents into constituents fail to account for important phrase alignments. As a consequence, straight-forward syntax-based mappings do not lead to better translations than unmotivated phrase mappings. This is a challenge for syntactic translation models.

It matters how phrases are extracted. The results suggest that choosing the right alignment heuristic is more important than which model is used to create the initial word alignments.

Current methods for learning phrase translation tables are based on a number of heuristics that have shown to be practical. However, there are many variations of these heuristics, and what performs best depends on training corpus and language pair. We foresee more research in this area: a more principled model of the phrase translation probability distribution that can be easily optimized for a novel corpus and language pair is desirable.