
Soft Features for Statistical Machine Translation of Spoken and Signed Languages

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von
Diplom-Informatiker
Daniel Stein
aus Düsseldorf

Berichter: Universitätsprofessor Dr.-Ing. Hermann Ney
Universitätsprofessor Andy Way, Ph.D.

Tag der mündlichen Prüfung: 18. Januar 2012

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online
verfügbar.

In Erinnerung an
Gisela Schischke

Sieh es mir nach, wenn ich weine.

Acknowledgements

I would like to express my deep gratitude towards Professor Hermann Ney, who granted me the possibility to, in his own words, build machines instead of pressing knobs and turning wheels. His proficiency in matters of science, his relentless search for the next step and his professionalism in human interaction has been and will continue to be most inspiring.

I would also like to sincerely thank co-supervisor Professor Andy Way for his kind hosting and general guidance during my stay in Dublin, as well as for his extensive feedback. I admit that he caught me a bit by surprise when he found my work truly interesting.

Many thanks to Prof. Jarke, Prof. Lakemeyer and PD Unger, who in one way or the other were always there during my whole academic career and did not hesitate to support me for the viva.

With profound admiration I would like to thank Davi(i)d “m-hm” Vilar for being the best Yoda I have ever had. His contribution to my structural thinking, scientific understanding and general way of programming is impossible to overestimate.

Ralf Schlüter, for sharing the student’s panopticon and for opening up the path to signal processing via brute-force: thank you Ralf, I owe you.

Special thanks to Gregor “Aaaaah, why?” Leusch and Arne “Makakosch” Mauser for occasionally allowing a short glimpse into their brain. A huge thank-you to Christoph “Schmitz” Schmidt for redirecting my foot steps towards a brighter future, Matthias “I lost my contact lense” Huck for helping me out in the educational stuff n’ more, and to Jan “history lessons” Bungeroth for saving me from those other guys. I would also like to thank Yannick “ouuh” Gweth, Christian “nach dem Motto” Plahl and David “undergrads always get this wrong” Rybach for their kind administration issue solving, and of course Kai “Protoss” Frantzen and Stefan “aumix” Koltermann. Thanks to Evgeny “mustache” Matusov for curing me of simply

running scripts, and thanks to Richard “ ” Zens for his inspiring code art. Thanks to Oliver “Hast Du keine Eval?” Bender, Arnaud “looks good” Dagnelies, Thomas “Exploitation” Deselaers, Philippe “DJ” Dreuw, Minwei “LM” Feng, Jens “Don’t panic” Forster, Saša “jaja, wart mal ab” Hasan, Carmen “Jovi” Heger, Stefan “Mal n’ bißchen nett hier” Hahn, Björn “guinea pigs” Hoffmeister, Sharam “Sidney” Khadivi, Patrick “Genieß es solange Du. . .” Lehnen, Jonas “Dagegen” Lööf, Amr “if you print it, don’t throw it away” Moussa, Saab “where’s the party?” Mansour, Markus “sorry ey” Nußbaum(-Thom), Maja “mumblemumble” Popović, Martin “I (uhm) have an idea” Ratajczak, Martin “Genau, hab ich mir nämlich auch gedacht” Sundermeyer, Simon “Bio-coffee” Wiesler, Jia “Petabyte” Xu, Morteza “ringringring” Zahedi, and Yuqi “hello” Zhang. With fondness I thank my diploma students Stephan “ja gut ok” Peitz, Markus “ey komm mal klar, Junge” Freitag and Jan-Thorsten “macht Sinn” Peter for taking over the curse, thus freeing me.

Thanks to Gisela and Katja for the chitchat and the always friendly Stundenzettel reminder.

Cheers to the Dublin City University, Ireland, the Deaf Sign Language Research Team, Aachen, Germany, and the Universidad Politécnica de Valencia, Spain.

Special thanks to you that you are reading this and wondering why you are not addressed personally. I have not forgotten you (but you knew this would happen, didn’t you?)

Last not least: thanks to my wife Julia for her understanding and her incredible support, and my daughter Ronja for checking the status of my experiments in the grid engine queue before rock n’ rolling into our lives.

Thanks, folks. It has been amazing.

Contents

1	Introduction	1
1.1	About this Document	1
1.1.1	(A Brief History of) Statistical Machine Translation	2
1.1.2	Soft Features for the Decoder	3
1.1.3	Non-Syntactic vs. Syntactic Features	4
1.1.4	Spoken, Written and Signed Languages	4
1.2	Related Work	5
1.3	Document Structure	7
1.4	Previously Published	7
2	Scientific Goals	11
3	Preliminaries	13
3.1	Basic Terminology	13
3.2	Bayes' Decision Rule for Statistical Machine Translation	14
3.2.1	Language Model	14
3.2.2	Translation Model and Alignments	16
3.3	Bilingual Phrase Pairs	17
3.3.1	Lexical Phrases	19
3.3.2	Synchronous Context Free Grammar	20
3.3.3	Hierarchical Phrases	20
3.4	Decoding in Hierarchical Machine Translation	22
3.5	Log-Linear Model	23
3.6	Optimization	24
3.6.1	Error Measures and Scores	25
3.6.2	Minimum Error Rate Training	25

4	Jane: Open Source Hierarchical Decoder	29
4.1	Implementation Overview	29
4.2	Core Functionality	30
4.2.1	Extraction	30
4.2.2	Generation	32
4.2.3	Optimization Methods	33
4.3	Advanced Models	34
4.3.1	Additional Reordering Models	34
4.3.2	Extended Lexicon Models	35
4.4	Extensibility	38
4.5	Comparison with Joshua	39
4.6	Conclusion	41
5	Soft Syntactic Features	43
5.1	Syntactic Parsing	44
5.2	Parse Matching	44
5.3	Soft Syntactic Labels	46
5.4	Soft String-To-Dependency	49
5.5	Experiments	54
5.6	Conclusion	56
6	Sign Language Corpus Analysis	61
6.1	Sign Language Grammar	61
6.1.1	Notation Systems	62
6.1.2	Components	63
6.1.3	Selected Phenomena in Sign Languages	64
6.2	Sign Language Corpora	66
6.2.1	Corpus RWTH-PHOENIX	66
6.2.2	Corpus NGT	72
6.2.3	Other Corpora	74
6.3	Usefulness of Sign Language Machine Translation	74
6.3.1	Sanity Check: Lower Case Translation	75
6.4	Conclusion	76
7	Sign Language Translation	77
7.1	Related work	78
7.2	System Description and Preliminaries	79
7.2.1	PBT: Phrase-based Translation	79
7.2.2	Jane: Hierarchical Phrase-based Translation	79
7.2.3	System Combination	79
7.2.4	Alignment Merging Strategies	79
7.2.5	Evaluation	82
7.3	Preparation of Sign Language Corpora	82
7.3.1	Translation Paradigm Comparison	82

7.3.2	Translation from a Morphologically Complex Spoken Language	83
7.3.3	Translating from Two Input Streams	85
7.4	Optimizing Sign Language Machine Translation	88
7.4.1	On the Choice of the Development Corpus	88
7.4.2	On the Choice of the Training Criterion	90
7.4.3	Linguistically Motivated Approaches	91
7.4.4	System Combination	91
7.5	Conclusion	94
8	Scientific Achievements	95
A	Curriculum vitae	99
B	Overview of the Corpora	101
B.1	NIST Chinese–English	101
B.2	GALE Arabic–English	101
B.3	QUAERO German–French	101
C	Sign Language Gloss Annotation Conventions	105

The goal of Machine Translation (MT) is to automatically translate a text from a source language correctly into a target language. In order to achieve this, one has to rely on human knowledge, collected by experts that are fluent in both languages. There are two distinct approaches as to where this knowledge should be applied. The *rule-based translation* approach makes use of hand-written rules for the actual language translation. The other approach, which is often referred to as *data-driven translation*, employs large collections of already translated material and tries to come up with its own, automatically derived rules between these languages.

Most data-driven systems tend to produce rules which contain little or no connection to common linguistic concepts. Hence, one might find it surprising that they often enough outperform rule-based approaches in international evaluations. Due to their success, these methods have been widely accepted as a mainstream approach over the last decade. Their strength lies in their generality, with many findings easily carrying over to other language pairs. Another benefit is simply a matter of speed. While every new domain in a rule-based system, even more so every language pair, needs to be defined anew by linguistic experts in a rather tedious task often exceeding years of work, a new statistical translation system can be trained within a few hours to days. The only limiting precondition is that sufficiently sized data collections can be accessed. However, more and more suitable data collections become available every year, now already exceeding ten million translated sentences for some language pairs like Chinese–English.

1.1 About this Document

This document presents and discusses various extensions to the data-driven MT approach. More specifically, it employs soft features within a statistical MT framework, on both spoken languages and sign languages. We will

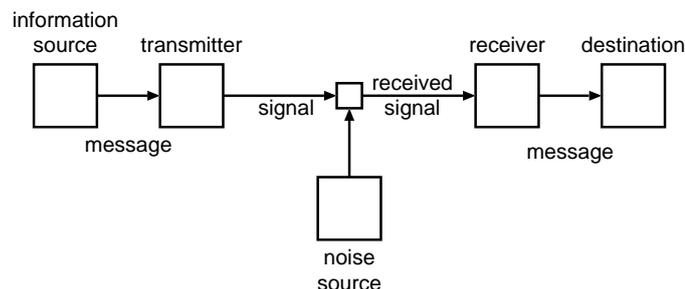


Figure 1.1: Schematic diagram of a general communication system

proceed to explain these individual terms below.

1.1.1 (A Brief History of) Statistical Machine Translation

Statistical Machine Translation (SMT) is probably the most prominent data-driven translation method. Its core idea is to treat the translation problem as a deciphering problem, a viewpoint that can be traced back to a comment by Warren Weaver [Weaver 55] on the works of Claude Shannon [Shannon 48]. Weaver suggested that translation might be a particularly promising application within the (then nascent) field of communication theory and its suggested statistical characteristics of the communication process (see Figure 1.1). Following this line of thought, a person transmitting some information might have encrypted it in a foreign language, which the receiver would then have to decrypt to obtain the original message. Even nowadays, the translation algorithm is often referred to as *decoder*, a convention that we will adopt throughout this work. Weaver noted that

[.] it is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the “Chinese code”.

Expectedly, the concept of a purely statistical approach to MT has not been without dispute in the scientific community. Noam Chomsky formulates in [Chomsky 68] a most famous remark:

Presumably, a complex of dispositions is a structure that can be represented as a set of probabilities for utterances in certain definable “circumstances” or “situations”. But it must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.

Many scientists in the SMT field feel that this influential pronouncement has considerably decelerated their early research. Today, Chomsky’s

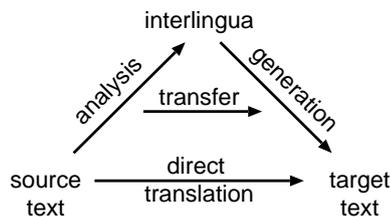


Figure 1.2: [Vauquois 68] Pyramid diagram of translation approaches

statement is generally considered to be refuted. One researcher notes in [Ney 03]:

By hindsight, one might try to come up with many possible explanations for [Chomsky’s] claims. But what remains in the end is the conclusion that they are nonsense and that Chomsky had too little knowledge of statistics.

The scientific rebound is generally attributed to the IBM Research Division. Many pioneering formula, findings and algorithms have been proposed by this group in the early 1990s (e.g. in [Brown & Cocke⁺ 90] and [Brown & Della Pietra⁺ 93]). The group concentrated their work on parliamentary debates from the Canadian Hansards. In Canada, where both French and English are official languages, all speeches given by politicians have to be translated into the other language by law. The fact that it is still custom to denote the source language as f (“French”) and the target language as e (“English”) can be seen as an indication of how influential these works have been.

In Section 3.2, we will review the mathematics of SMT.

1.1.2 Soft Features for the Decoder

In principle, an SMT decoder translates a source text by assigning a probability to every target text that it can possibly come up with. The target text having the highest probability is put forth as the best translation. The probabilities are typically computed with a large variety of different models that are called *feature functions*. Most feature functions evaluate dependencies between the given source text and each proposed target text. Some only work on one language part, for example by trying to measure the fluency of the target text. Feature models that take both languages into account are called *translation models*, while models working on only one language are called *language models*. Note that, while the term “language model” is commonly only applied to probability distributions over the target language, we will extend this term to every monolingual function to facilitate upcoming

definitions, as we believe that the concrete meaning should be obvious from the context.

In the following chapters, we will present and review various translation models and language models. Common to them is that they will never restrict possible translations by assigning zero probabilities. Moreover, the decoder is in principle allowed to ignore the whole feature function. We therefore denote these features to be *soft features*.

1.1.3 Non-Syntactic vs. Syntactic Features

It was stated earlier that data-driven approaches typically neither take linguistic considerations into account nor produce translation rules that seem to have a strong linguistic justification. Following the classification scheme of [Vauquois 68] (see Figure 1.2), we denote a translation without any linguistic or semantic analysis a *direct translation*. Human translation presumably tries to grasp the semantic meaning of a sentence first by analyzing its meaning, and then generates the target language from an *interlingua*. This approach, while certainly more appealing from an intuitive point of view, seems quite problematic to accomplish. All three steps, namely analysis, semantic representation and generation, are in themselves already challenging. If they are concatenated, the overall pipeline seems even more ambitious.

Nevertheless, already some intermediate approaches, halfway towards the top of the pyramid, could improve the feature functions, which would then guide the decoding process in a better direction. By relying on e.g. syntactic analysis of the languages, they could help to *transfer* meta-language information into the target text. In Chapter 5, we will present and review three additional feature models that take linguistic analysis by means of automatic language parsers into account.

One might argue whether a MT system working with automatic linguistic analysis tools is still a purely data-driven approach. Perhaps the notion *hybrid approach* would be more appropriate (cf. [Groves & Way 05]).

1.1.4 Spoken, Written and Signed Languages

We follow the notation of [Bellugi & Fischer 72] and refer to languages which can be acoustically conveyed with sound patterns as *spoken languages*. They are to be distinguished from *signed languages*, which instead transmit visual sign patterns through body language and manual communication. Consequently, we refrain from the usage of the notion *written language* (e.g. [d'Armond L. Speers 02]) to set the written form of a spoken language apart from a signed language, since we will also deal with written transcriptions of signed languages throughout this thesis.

Signed languages are the primal means of communication for most deaf and many hard-of-hearing persons. If evolved naturally, almost all signed

languages differ at great length from spoken languages, by having both unique grammar and vocabulary, and by being able to convey meaning simultaneously on different communication channels: manual information like hand shape, orientation, position and movement of the hands, and non-manual information like body posture and facial expression. Most signed languages are unlimited in their expressiveness. With the use of the parallel channels, [Bellugi & Fischer 72] even suggest that American Sign Language (ASL) conveys some information faster than spoken English. For SMT, signed languages are an interesting niche area because of their non-sequential nature, and because the data collections that can be employed in an SMT framework are typically scarce.

In Chapter 6, we will analyze data collections for several sign languages, and will present several suitably tailored methods for sign language MT in Chapter 7.

1.2 Related Work

This section lists the scientific work that is most closely related to the methods proposed in the following chapter. For readability purposes, we split the citations into the relevant parts.

Jane: Open Source Hierarchical Decoder (Chapter 4)

In this chapter, we present and discuss Jane, an open source translation toolkit which was developed as part of this thesis. Jane implements many previous ideas developed both at RWTH Aachen University and other groups. As we go over the features of the system we will provide the corresponding references, and only list comparable, full-sized toolkits here. Jane is not the first system of its kind, although it provides some unique features. There are other open source hierarchical decoders available.

The Syntax Augmented Machine Translation (SAMT) decoder was developed by Carnegie Mellon University, Pittsburgh, USA, and released in [Zollmann & Venugopal 06]. The original version is not maintained any more and we had problems getting it to work on big corpora. Another decoder is Joshua [Li & Callison-Burch⁺ 09], a joint piece of work of several departments hosted by the Johns Hopkins University, Baltimore, USA. This project is the most similar to our own, but both were developed independently and each one has some unique features. A more in-depth comparison between these two systems is included in Section 4.5. Lately, the de-facto standard phrase-based translation decoder Moses [Koehn & Hoang⁺ 07], another joint effort hosted by the University of Edinburgh, Scotland, has been extended to support hierarchical translation.

Soft Syntactic Features (Chapter 5)

In this chapter, we incorporate syntactic knowledge into a SMT framework. [Yamada & Knight 01] was one of the first works in this area, al-

though the performance was not on par with other state-of-the-art approaches at that time. Further development in this direction achieved competitive results, as can be seen in [DeNeefe & Knight⁺ 07] and later publications by the same group.

In contrast to these works, which propose new models centered around syntactic information, we focus mainly on methods that can be easily incorporated into an existing hierarchical system. In this work, we employ soft syntactic features as in [Vilar & Stein⁺ 08a]. These features measure to what extent a phrase corresponds to a valid syntactic structure of a given parse tree. In addition, we include a dependency language model in a string-to-dependency model in the spirit of [Shen & Xu⁺ 08]. We also derive soft syntactic labels as in [Venugopal & Zollmann⁺ 09], where the generic non-terminal of the hierarchical system is replaced by a syntactic label. [Almaghout & Jiang⁺ 10] have also extended their decoder in a similar way, using a Combinatory Categorical Grammar to derive their labels.

[Marton & Resnik 08, Chiang & Knight⁺ 09, Chiang 10] are working in similar directions, but create a rather large quantity of features.

Sign Languages (Chapter 6)

In this chapter, we offer some findings in the field of sign language corpus creation. Recently, a couple of other sign language data collections have been created. Based on their purpose, some of them have only limited usability to data-driven natural language processing techniques. Listed below are some of the larger efforts for European sign languages, which we will discuss more in detail in the chapter itself.

Similar to our corpus presented in [Bungeroth & Stein⁺ 06], where we presented a sign language corpus for German and German Sign Language in the domain of weather forecast, other groups have started to build corpora in the same domain: [Bertoldi & Tiotto⁺ 10] for Italian and Italian Sign Language, and [Massó & Badia 10] for Catalan and Catalan Sign Language.

Other corpora include: [Kanis & Zahradil⁺ 06], a corpus for Czech and Signed Czech. Its domain is taken from transcribed train timetable dialogues and then translated by human experts. [Bungeroth & Stein⁺ 08] is a corpus for English, German, Irish Sign Language, German Sign Language and South African Sign Language in the domain of the Air Travel Information System (ATIS). With roughly 600 parallel sentences in total, it is small in size. However, being a multilingual data selection, it enables direct translation between sign languages. [Crasborn & van der Kooij⁺ 04] is a corpus for Swedish Sign Language, British Sign Language and Sign Language of the Netherlands. However, their broad domain of children’s fairy tales as well as poetry make them rather unsuitable for statistical methods. Another obstacle is the intensive usage of signed classifiers because of the rather visual topics.

Sign Language Translation (Chapter 7)

In this chapter, we deal with sign language translation by means of SMT.

One of the first ideas made in this field was given by [Bauer & Kraiss 01, Sáfár & Marshall 01, Huenerfauth 03], but these papers do not offer much experimental results. Recent works in this area include [Morrissey 08], which is an in-depth investigation of corpus-based methods for data-driven sign language translation from English to Irish Sign Language. A system for the language pair Chinese and Taiwanese sign language is presented in [Chiu & Wu⁺ 07]. They show that their optimizing method surpasses IBM model 2. [Kanis & Müller 09] report quite high performance on a Czech to Signed Czech task, and we will review their findings more closely. [Massó & Badia 10] use factored models on a standard phrase-based system for Spanish to Spanish Sign Language.

1.3 Document Structure

The main chapters of this thesis are organized as follows. In Chapter 2, we will present the scientific achievements presented in this thesis. Chapter 3 gives a brief introduction into the relevant mathematical foundations and techniques used in SMT. Chapter 4 presents the decoder that was developed in the course of this thesis, along with the discussion of our implementation decisions. Chapter 5 presents and reviews three linguistically motivated models and compares how they interact with each other. In Chapter 6, we introduce sign language corpora, discuss their annotation and characteristics. The special demands for translation will be analyzed in Chapter 7. Finally, we conclude this work and summarize our findings in Chapter 8.

1.4 Previously Published

During the course of this thesis, the following scientific publications have been successfully submitted to peer-reviewed conferences and journals:

- International Evaluation Campaigns
 - [Vilar & Stein⁺ 08b] The RWTH Machine Translation System for IWSLT 2008
 - [Popović & Vilar⁺ 09] The RWTH Machine Translation System for WMT 2009
 - [Heger & Wuebker⁺ 10] The RWTH Aachen Machine Translation System for WMT 2010
 - [Huck & Wuebker⁺ 11] The RWTH Aachen Machine Translation System for WMT 2011
- JANE and Syntactic Enhancements

- [Popović & Stein⁺ 06] Statistical Machine Translation of German Compound Words
 - [Vilar & Stein⁺ 08a] Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation
 - [Vilar & Stein⁺ 10a] Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models
 - [Stein & Peitz⁺ 10] A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation
 - [Vilar & Stein⁺ 10b] If I Only Had a Parser: Poor Man’s Syntax for Hierarchical Machine Translation
 - [Stein & Vilar⁺ 11a] A Guide to Jane, an Open Source Hierarchical Translation Toolkit (article)
 - [Stein & Vilar⁺ 11b] Soft Syntax Features and Other Extensions for Hierarchical SMT
 - [Peter & Huck⁺ 11] Soft String-to-Dependency Hierarchical Machine Translation
 - [Vilar & Stein⁺ 12] Jane: An Advanced Freely-Available Hierarchical Machine Translation Toolkit (article, to appear)
- Sign Language Corpora
 - [Bungeroth & Stein⁺ 06] A German Sign Language Corpus of the Domain Weather Report
 - [Bungeroth & Stein⁺ 08] The ATIS Sign Language Corpus
 - [Ormel & Crasborn⁺ 10] Glossing a Multi-purpose Sign Language Corpus
 - [Forster & Stein⁺ 10] Best Practice for Sign Language Data Collections Regarding the Needs of Data-Driven Recognition and Translation
- Sign Language Translation
 - [Stein & Bungeroth⁺ 06] Morpho-Syntax Based Statistical Methods for Sign Language Translation
 - [Dreuw & Stein⁺ 07] Enhancing a Sign Language Translation System with Vision-Based Features (extended abstract)
 - [Morrissey & Way⁺ 07] Towards a Hybrid Data-Driven MT System for Sign Languages
 - [Stein & Dreuw⁺ 07] Hand in Hand: Automatic Sign Language to Speech Translation
 - [Dreuw & Stein⁺ 08] Spoken Language Processing Techniques for Sign Language Recognition and Translation

- [Dreuw & Stein⁺ 09] Enhancing a Sign Language Translation System with Vision-Based Features (article)
 - [Dreuw & Forster⁺ 10a] SignSpeak – Understanding, Recognition, and Translation of Sign Languages
 - [Stein & Forster⁺ 10] Analysis of the German Sign Language Weather Forecast Corpus
 - [Stein & Schmidt⁺ 10] Sign Language Machine Translation Overkill
 - [Stein & Schmidt⁺ 12] Analysis, Preparation, and Optimization of Statistical Sign Language Machine Translation (article, to appear)
- Other (minor contributions)
 - [D’Haro & San-Segundo⁺ 08] Language Model Adaptation For a Speech to Sign Language Translation System Using Web Frequencies and a Map Framework
 - [Huck & Vilar⁺ 11a] Advancements in Arabic-to-English Hierarchical Machine Translation
 - [Huck & Vilar⁺ 11b] Lightly-Supervised Training for Hierarchical Phrase-Based Machine Translation

Scientific Goals

In this thesis we are going to pursue the following scientific goals:

- We will establish a new hierarchical MT decoder that achieves state-of-the-art performance. The toolkit will be employed on several large-scale corpora for several language pairs. We will conduct comparison experiments to the conventional phrase-based translation approach, and we will further compare our decoder to a similar decoder called Joshua [Li & Callison-Burch⁺ 09], developed as joint work at Johns Hopkins University.
- We will present an easy method for marking the syntactical soundness of a phrase and analyze its impact on several language pairs.
- For the soft preference grammar [Venugopal & Zollmann⁺ 09], we will describe an approach to label phrases which do not match the yield of a parse node. Thus, we are able to greatly reduce the number of necessary labels.
- We will extend the string-to-dependency approach [Shen & Xu⁺ 08] so that the phrase table is not reduced to phrases which match certain dependency conditions. We show that this does not reduce the impact of the model, and moreover this enables other models to behave normally.
- All three syntactic methods have been shown to improve the translation quality individually. In this work, we will apply all of them simultaneously for the first time, and compare their individual performance as well as their combination ability on a common baseline.
- We will give a detailed overview of existing sign language data collections. We further introduce the Corpus-NGT for Spoken Dutch and

Sign Language of the Netherlands, which is broad domain and has a rich annotation of the parallel communication channels.

- We will offer findings on how to prepare a sign language MT system, by adapting the phrase table extraction procedure to the needs of these language pairs, and by proper preprocessing of the source language based on its modality.
- We analyze statistical MT for scarce resources in detail, by examining to what degree additional models derived for medium-scale and large-scale corpora can be applied on sign language data collections. We introduce several methods which are suitably tailored and which can be applied to other under-resourced language pairs.

In this chapter, we will review the underlying principles of the methods described in this work, as is standard in our field. The topics have been previously presented by several authors. Hence, we restrict the following sections only to the most basic concepts needed to understand our own experiments, referring the interested reader to the given papers.

This chapter is organized as follows: after defining some basic terminology in Section 3.1, we review the language model and the translation model in Section 3.2. Defining bilingual phrases and synchronous context free grammars in Section 3.3 enables us to briefly discuss the principal workflow of hierarchical decoding in Section 3.4. In Section 3.5, we will review the log-linear model, so that we can incorporate a larger number of models into the translation framework. In order to judge the quality of a given hypothesis, we review common error measures and scores in Section 3.6, and will also discuss the optimization of the feature function scaling factors.

3.1 Basic Terminology

As already mentioned briefly in Section 1.1.1, we treat the translation process as a deciphering problem and thus call a MT system a *decoder*. To differentiate between human translation and MT, we call a human-translated document a *reference*, and an automatically decoded document a *hypothesis*. A collection of documents and their references is called a *corpus*.

To test the generalization abilities of our models, we split some portions from our corpus. The largest part is used as *training material* for the models. The models are then further optimized on a withheld portion called the *development set*, and finally tested on a *test set*.

Note that we simplify some of the following concepts to the translation of sentences rather than full documents, and indeed will not go beyond sentence-wise translation within this work. A source sentence consisting of

J words is denoted as $f_1^J = f_1 f_2 \dots f_J$, and a target sentence consisting of I words is denoted as $e_1^I = e_1 e_2 \dots e_I$. A succession of n words will be denoted as an n -gram. For $n = 1, \dots, 5$, n -grams are called *unigrams*, *bigrams*, *trigrams*, *four-grams*, *five-grams*. The predecessor words of a word in a sentence are called its *history*.

3.2 Bayes' Decision Rule for Statistical Machine Translation

In SMT, we theoretically translate a source sentence by computing the a-posteriori probability of all target sentences that we can possibly come up with. The sentence that maximizes this probability is selected as our hypothesis. We can formulate the decoding process as:

$$f_1^J \rightarrow \hat{e}_1^I(f_1^J) = \operatorname{argmax}_{e_1^I} \{Pr(e_1^I | f_1^J)\} . \quad (3.1)$$

In a first attempt to model $Pr(e_1^I | f_1^J)$ in Equation 3.1, we could apply Bayes' theorem to split the probability into more convenient terms:

$$f_1^J \rightarrow \hat{e}_1^I(f_1^J) = \operatorname{argmax}_{e_1^I} \left\{ \frac{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)}{Pr(f_1^J)} \right\} \quad (3.2)$$

$$= \operatorname{argmax}_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \quad (3.3)$$

The denominator of Equation 3.2 does not influence the argumentum maximandi and can be omitted. We thus face two sub-models: the *language model* $Pr(e_1^I)$ and the *translation model* $Pr(f_1^J | e_1^I)$. We will discuss both models below.

3.2.1 Language Model

A language model (LM) is the a-priori probability distribution $Pr(e_1^I)$ over a string e_1^I . Roughly speaking, it measures to what extent a sentence could belong to our target language, and hopefully penalizes ill-formed sentences. For every word e_i , we theoretically have to consider its full history:

$$Pr(e_1^I) = \prod_{i=1}^I Pr(e_i | e_1^{i-1}) . \quad (3.4)$$

We typically restrict the LM in such a way that it only considers n -grams. The history is then limited to $n - 1$ words $h_i = e_{i-n+1}^{i-1}$, with common values for n ranging between 3 and 7. This limitation mostly has computational

and modelling considerations, and it seems to be a reasonable assumption that not all words at the end of a sentence will depend on the words at the beginning. Still, from a linguistic point this constraint is unsatisfactory, since some long-range dependencies are to be expected within a natural language sentence. For example, in German the prefix of a verb can be split and appear at a completely different sentence position to the verb stem. If the distance is too large, the LM will never evaluate both words simultaneously. We will review this problem in Section 5.4.

An estimate of the quality of a LM is given by computing the *perplexity* on a development set. Roughly speaking, the perplexity measures how “surprised” the algorithm is to see a word e , given its history h . The perplexity of a language model is commonly defined as:

$$PP = \Pr(e_1^T)^{-\frac{1}{T}} \quad (3.5)$$

Maximum Likelihood Estimation and its Limitation

In the process of training a language model, we select the parameters of a probability model so that its probability is maximized on a training document. We call this approach the *maximum likelihood criterion*.

Let our training material consist of a document w_1^N with N words, and let (w_n, h_n) be its n-grams. Let $p_\theta(w_n|h_n)$ be a probability model with a set of free parameters θ . Then, our maximum likelihood criterion can be formulated as:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \left\{ \prod_{n=1}^N p_\theta(w_n|h_n) \right\} \quad (3.6)$$

If we compute the optimum for the maximum likelihood criterion as in Equation 3.6, we can easily ensure that the probabilities are normalized for all histories. The maximum likelihood estimates result in relative frequencies of the n-gram counts:

$$p_\theta(w|h) = \frac{N(h, w)}{\sum_{w'} N(h, w')} \quad (3.7)$$

$$\text{with } N(h, w) := \sum_{n:(h,w)=(h_n,e_n)} 1 \quad (3.8)$$

However, while Equation 3.7 is the probability model with the highest probability on our training data, it might not generalize well on unseen data. A problem arises whenever we encounter an n-gram that we have not seen in training, since relative frequencies will assign a zero probability to

this n-gram. For real-life data, this will happen considerably often. In a document consisting of, say, $10 \cdot 10^6$ words and a vocabulary size of $2 \cdot 10^4$ words, there are $4 \cdot 10^8$ possible bigrams, of which we can only encounter a maximum of 2.5%. For trigrams, a mere $1.25 \cdot 10^{-4}\%$ of all possible trigrams could be encountered in the text. Such a model will be quite inflexible, so we rather might want to reserve some low probabilities to unseen n-grams. This technique is called *smoothing* (e.g. [Ney & Essen⁺ 94]).

Handling of Unseen Events with Smoothing

A language model based on relative frequencies will assign zero probabilities to each n-gram not encountered in the training material, which is an unwanted effect. We therefore shift some probability mass from the relative frequencies and assign very low, but non-zero, probabilities to the unseen n-grams. If we subtract a linear portion of the relative frequencies, this is called *linear discounting*. If we subtract absolute values from the n-gram counts before the computation of the relative frequencies, this is called *absolute discounting*.

In our work, we use the *modified Kneser-Ney* smoothing. It is an absolute discounting method that differentiates between n-grams that appear once, twice, and more than twice. For more information about the method, we refer to [Ney & Essen⁺ 94].

3.2.2 Translation Model and Alignments

When applying Bayes' decision theorem on the a-posteriori probability in Equation 3.3, we have seen that we obtain another sub-term apart from the language model: the translation model $Pr(f_1^J | e_1^I)$. The translation model assigns a probability to one sentence f_1^J of being the translation of another sentence e_1^I . This task seems harder to compute than a language model, since it involves two languages at once. However, it seems safe to assume that a given word in the source sentence does not correspond to each and every word in the target sentence. As is common practice, we therefore introduce a hidden variable called *alignment* \mathcal{A} . An alignment is a set of postulated word-to-word correspondences called *alignment points*, and we train them as part of our overall training process. To model words that might not be translated into the other language at all, we also introduce the *empty word*.

Without loss of generality, we can now reformulate the translation model as:

$$Pr(f_1^J | e_1^I) = \sum_{\mathcal{A}} Pr(f_1^J, \mathcal{A} | e_1^I) \quad (3.9)$$

$$Pr(f_1^J, \mathcal{A} | e_1^I) = Pr(J | e_1^I) \cdot Pr(f_1^J, \mathcal{A} | J, e_1^I) \quad (3.10)$$

$$= Pr(J | e_1^I) \cdot Pr(\mathcal{A} | J, e_1^I) \cdot Pr(f_1^J | \mathcal{A}, J, e_1^I). \quad (3.11)$$

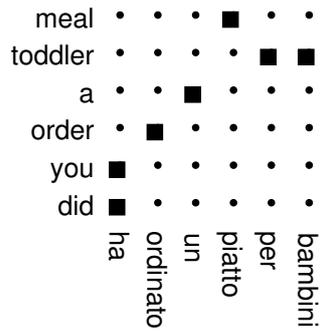
Once more, we arrive at smaller subproblems that we are able to model individually. The new models are called the *length model* $Pr(J | e_1^I)$, the *alignment model* $Pr(\mathcal{A} | J, e_1^I)$ and the *lexicon model* $Pr(f_1^J | \mathcal{A}, J, e_1^I)$. In the pioneering work of [Brown & Della Pietra⁺ 93], a succession of model assumptions for these subproblems was introduced. They are called *IBM models*, and are designed to build upon each other, i.e. IBM Model 1 will have the most simplifying assumptions and merely produces some early estimates, serving as initialization for later models. The estimates can be improved in IBM Model 2, passed on to IBM Model 3 and so on. We will continue to briefly describe the core idea of these models, but refer the reader to the aforementioned paper for further reading.

IBM Model 1 and 2 are *zero-order* models, since they do not take any decision from surrounding words into account. The alignments are limited to single word correspondences that are source position-dependent, i.e. $\mathcal{A} := a_1^J = a_1 \dots a_j \dots a_J$, an assumption which reduces the number of possible alignments from $2^{I \cdot J}$ to I^J . IBM Model 2 is similar, but extends its predecessor model with a source position-dependent alignment model. In this thesis, we employ an extension of IBM Model 2 that is based on a Hidden Markov Model (HMM). This variant is a first-order model since it also takes the preceding alignment position into account [Vogel & Ney⁺ 96].

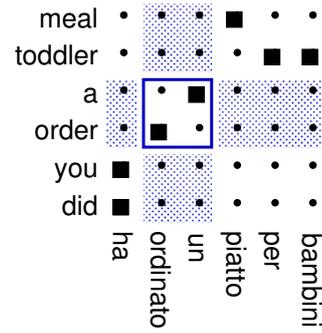
Beyond IBM Model 2, the word-to-word translation is extended by introducing the *word fertility* ϕ_j . With this new concept, we can allow for certain words in one language to produce more than one word in the other. The alignment model is computed into the other direction, and the higher-range IBM Models mainly model the distribution of the alignment probability further. It is interesting to note that, based on their constructions, the IBM Models will lead to direction-dependent alignments, i.e. the alignments will differ when we interchange source and target. In practice, alignments for both language directions will be computed and then merged with some heuristic.

3.3 Bilingual Phrase Pairs

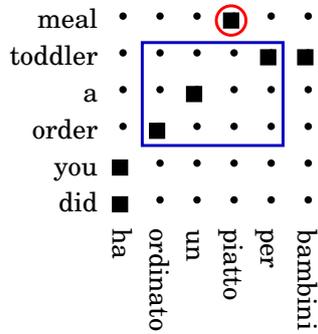
We introduced the translation models as given in [Brown & Della Pietra⁺ 93] to familiarize the reader with the concepts of alignments. While the alignments of a corpus are typically still computed with the IBM Models, the actual translation method of current state-of-the-art decoders differs. Even



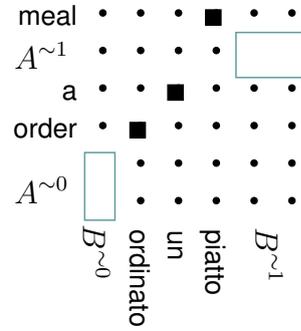
(a) Alignment example for an Italian sentence and its English translation. Each circle in the matrix represents a possible word-to-word correspondence, each square represents an actual assignment.



(b) Example of a valid lexical phrase defined by Equation 3.12



(c) Example of a phrase considered invalid by Equation 3.12



(d) Example of a hierarchical phrase as defined by Equation 3.15

Figure 3.1: Visualisation of an alignment, of valid and invalid lexical phrases and hierarchical phrases

the more sophisticated IBM Models conduct the actual translation only on a word basis, i.e. $p(f_j|e_i)$. With this approach, local context information cannot be taken into account. This is a large source of possible errors since the language model is the only method that will check several consecutive words and their relation towards each other.

Consider for example the German word “Kater”, a homonym which can mean both “cat” or “hangover” in English. If through context we can derive that the “Kater” is purring (and given that the narrator is not in a particularly metaphorical mood), we might conclude that the German sentence is talking about animals rather than delayed, alcohol-induced headache. In this section, we are looking at concepts that attempt to find larger blocks of source and target words within the sentence.

3.3.1 Lexical Phrases

In [Och & Tillmann⁺ 99, Zens & Och⁺ 02, Koehn & Och⁺ 03], the word-based translation approach was extended towards the more versatile approach of Phrase-based Translation (PBT). A valid phrase pair consists of contiguous words in both languages that share at least one alignment point. Furthermore, we do not allow alignment points from outside one phrase to be inside the phrase of the other language. The set \mathcal{BP} of valid bilingual phrases is defined as:

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, \mathcal{A}) := \{ \langle f_{j_1}^{j_2}, e_{i_1}^{i_2} \rangle \mid j_1, j_2, i_1, i_2 \quad \text{so that} \\ \forall (j, i) \in \mathcal{A} : (j_1 \leq j \leq j_2 \Leftrightarrow i_1 \leq i \leq i_2) \\ \wedge \exists (j, i) \in \mathcal{A} : (j_1 \leq j \leq j_2 \wedge i_1 \leq i \leq i_2) \}. \end{aligned} \tag{3.12}$$

For a sample alignment as in Figure 3.1(a), one example for a valid phrase is given in Figure 3.1(b) since no alignment position violates Equation 3.12, and 3.1(c) represents an invalid phrase since “piatto” is within the Italian phrase but aligned to “meal” which is outside the English phrase. We refer to the valid phrases as *lexical phrases*. With such phrases, we will be able to take local context information into account.

While this is certainly a step into the right direction, there are many phenomena in natural languages that still render this approach unsatisfactory. Take for example the German phrase “Ich stimme dem Antrag zu”, which is translated into English as “I agree with this proposal”. Problematic here is that the German verb “zustimmen”, meaning “agree”, is split into the stem verb “stimmen” and its prefix “zu”. The verb thus frames the object of the sentence, “dem Antrag”, and the smallest phrase that contains both parts of the verbs would be $\langle \text{stimme dem Antrag zu}, \text{agree to this proposal} \rangle$. This is of course a valid translation, but if this is the only phrase pair that

includes the verb, then Germans could never agree to anything else than this particular proposal.

If we allow phrases to contain gaps, we can model this example better. Extending a phrase with gaps that serve as place-holders for smaller phrases requires some formal definition first, and for this we start with a Context Free Grammar (CFG) as defined in [Chomsky 56].

3.3.2 Synchronous Context Free Grammar

Standardally, we can define a context free grammar as follows: let Σ be a set of *terminals* (i.e. words), N be a finite set of *non-terminals* that are disjoint from Σ , and let $S \in N$ be its start symbol. Let R be a set of production rules with $R \subset N \times (\Sigma \cup N)^*$. We can then denote our CFG with the 4-tuple (N, Σ, R, S) .

We use the symbol “ \rightarrow ” within a *rule* and the symbol “ \Rightarrow ” within a single *derivation*. For example, let A be a non-terminal $A \in N$ and let α, β and γ be arbitrary succession of terminal and non-terminal symbols $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$. Then, a rule r could look like $r = A \rightarrow \beta \in R$. Furthermore, we could derive $\alpha\beta\gamma$ from $\alpha A \gamma$ by using rule r and would denote this as $\alpha A \gamma \Rightarrow \alpha\beta\gamma$. The reflexive and transitive closure of \Rightarrow is denoted as \Rightarrow^* .

We can use CFGs to produce monolingual strings with gaps in between that can be filled with smaller phrases. For translation task at hand, however, we need to model two languages at once. For this, we make use of a Synchronous Context Free Grammar (SCFG), an extension to a CFG presented in [Lewis II & Stearns 68]. The grammar was originally intended to formalize a compiler translation, but it can be applied to MT rather easily.

For the target side, we introduce an additional alphabet Γ , and further distinguish between the source non-terminal alphabet N_f and the target non-terminal alphabet N_e . The right hand side of the rules is a 3-tuple (α, β, \sim) with $\alpha \in (\Sigma \cup N_f)^*$ and $\beta \in (\Gamma \cup N_e)^*$. The symbol \sim denotes a one-to-one correspondence between the non-terminals of the string over the alphabet Σ and the string over the alphabet Γ . A derivation is now defined over pairs of strings, where each non-terminal substitution occurs in a synchronous way on both strings, as governed by the \sim correspondence.

With the SCFG, we now have all the tools ready to formalize phrase pairs with gaps, which are commonly called *hierarchical phrases*.

3.3.3 Hierarchical Phrases

A hierarchical phrase is a lexical phrase where smaller lexical phrases contained therein have been cut out and replaced by a place-holding variable [Chiang 05]. We denote \mathcal{F} as the alphabet in the source language and \mathcal{E} as the alphabet in the target language. N_f and N_e will denote the sets of

non-terminals that are disjoint with both \mathcal{F} and \mathcal{E} . Note that \mathcal{F} and \mathcal{E} are not necessarily disjoint. For example, names of certain places or persons are likely to be similar for many languages.

A hierarchical phrase then has the form

$$(A, B) \rightarrow \langle \alpha, \beta, \sim \rangle \quad (3.13)$$

with $A \in N_f$, $B \in N_e$, $\alpha \in (\mathcal{F} \cup N_f)^+$, $\beta \in (\mathcal{E} \cup N_e)^+$. Further, \sim is a one-to-one relation between the non-terminals in α and β , in which case we enforce the non-terminals to consist of the same symbol.

The example shown in Figure 3.1(d) would be denoted as

$$(A, B) \rightarrow \langle A \text{ order a } A \text{ meal, } B \text{ ordinato } B \text{ per bambini, } \{(0, 0), (3, 2)\} \rangle,$$

where the indexes in the definition of \sim refer to positions in the source and target phrases, i.e. (0,0) means that the index of the non-terminal A is the first symbol in both phrases, while the non-terminal B is the fourth symbol in the Italian phrase and the third in the English phrase. We will write the rules in a more compact notation by specifying the \sim relation between the non terminals directly in the right-hand side of the rule, as a super index of the non-terminals. The previous rule will be written as

$$(A, B) \rightarrow \langle A^{\sim 0} \text{ order a } A^{\sim 1} \text{ meal, } B^{\sim 0} \text{ ordinato un piatto } B^{\sim 1} \rangle.$$

Non-terminals having the same super index are bound via the \sim relation.

We denote the set of hierarchical rules that contain n non-terminals in the source and target string with \mathcal{H}_n . \mathcal{H}_0 is defined directly, while \mathcal{H}_n is defined recursively:

$$\begin{aligned} \mathcal{H}_0(f_1^J, e_1^I, \mathcal{A}) = & \left\{ (A, B) \rightarrow \langle f_{j_1}^{j_2}, e_{i_1}^{i_2} \rangle \mid j_1, j_2, i_1, i_2 \quad \text{so that} \right. \\ & \forall (j, i) \in \mathcal{A} : (j_1 \leq j \leq j_2 \Leftrightarrow i_1 \leq i \leq i_2) \\ & \left. \wedge \exists (j, i) \in \mathcal{A} : (j_1 \leq j \leq j_2 \wedge i_1 \leq i \leq i_2) \right\}. \end{aligned} \quad (3.14)$$

Let α and β be strings with possible gaps in the source language, i.e. $\alpha, \beta \in (\mathcal{F} \cup N_f)^*$, and let γ and δ be strings with possible gaps in the target language, i.e. $\delta, \gamma \in (\mathcal{E} \cup N_e)^*$. Then, the recursive definition for \mathcal{H}_n is

$$\begin{aligned} \mathcal{H}_n(f_1^J, e_1^I, \mathcal{A}) = & \left\{ (A, B) \rightarrow \langle \alpha A^{\sim n} \beta, \delta B^{\sim n} \gamma \rangle \mid j_1, j_2, i_1, i_2 : j_1 \leq j_2 \wedge i_1 \leq i_2 \right. \\ & \left((A, B) \rightarrow \langle \alpha f_{j_1}^{j_2} \beta, \delta e_{i_1}^{i_2} \gamma \rangle \in \mathcal{H}_{n-1}(f_1^J, e_1^I, \mathcal{A}) \right. \\ & \left. \wedge (A, B) \rightarrow \langle f_{j_1}^{j_2}, e_{i_1}^{i_2} \rangle \in \mathcal{H}_0(f_1^J, e_1^I, \mathcal{A}) \right) \left. \right\}. \end{aligned} \quad (3.15)$$

The set of hierarchical phrase pairs extracted from a corpus is the union of the hierarchical phrases extracted from each of its sentence pairs. We will denote this set as \mathcal{H} . As is common practice, n is set to 2, which simplifies the decoder implementation [Chiang 05].

We have now arrived at the quite versatile concept of phrases that can be filled with other phrases. Note that we are also able to model reordering of larger chunks, e.g. whenever the order of the super index is reversed for two non-terminals. However, before we review the actual decoding step, let us step back from the mathematics of these phrases and look at them again from an intuitive point of view.

First, we note that the indexes which limit the lexical phrases are not linguistically justified boundaries. They are derived from automatically generated word alignments, and while some of them might coincide with e.g. noun phrases, the large proportion will seem arbitrary with respect to common grammatical concepts. Moreover, if we encounter a sentence pair with a monotone alignment, we will extract a huge number of bilingual phrases, but we probably will not need most of them. Maybe we should penalize those phrases that do not correspond to meaningful sub-phrases, an approach that we will elaborate more in Section 5.2.

A second observation is that the non-terminals are generic and are treated as equal for every other phrase. However, it might be a bad idea to fill a gap derived from a verb phrase with a noun phrase during decoding. In Section 5.3, we will review this problem more in detail.

3.4 Decoding in Hierarchical Machine Translation

In this section, we will describe the decoding step of the Hierarchical Phrase-based Machine Translation (HPBT), i.e. the search for the best translation by means of hierarchical phrases. It is not the main focus of this work, which is why we will keep this section rather short.

The hierarchical phrases are based on SCFGs (see Section 3.3.2), and we employ a monolingual parser that tries to generate our source sentence based on \mathcal{H} , and generate the target sentence afterwards. More precisely, for each possible derivation in the source language we simultaneously build a derivation tree in the target language, and the yield of that tree will be one possible hypothesis. For the parsing of the source sentence, we can make use of the well-known Cocke-Younger-Kasami (CYK) algorithm [Cocke 69, Younger 67, Kasami 65] as a starting point, but use the refined version Cocke-Younger-Kasami⁺ (CYK⁺) [Chappelier & Rajman 98] since it does not require the grammar to be in Chomsky normal form.

In order to produce translations, we need to make some adjustments. Both algorithms are directed at parsing, i.e. they decide whether a string can be produced by a given grammar, which is not exactly what we are looking

for in the translation task. Even if we cannot parse the entire source string, e.g. because we encounter previously unseen words like city names, we still want the best possible translation that we can derive. It should also be noted that some models are computed beyond the level of one hierarchical phrase. For example, the language model works on $n - 1$ predecessor words for each word in the target sentence, and it happens quite frequently that these words are contained within another rule derived elsewhere, or are even not yet translated at all. For an in-depth comparison of various decoding approaches to hierarchical machine translation, we refer the reader to [Vilar 12].

3.5 Log-Linear Model

In Equation 3.3, we split the a-posteriori translation probability into a language model and a translation model. While mathematically correct, this approach suffers from the inconvenience that we cannot easily add more than these two knowledge-sources into the equation. In [Och & Ney 02], the mathematical foundation of SMT was extended by a *log-linear model*, where we model the a-posteriori probability directly:

$$p(e_1^I | f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, \tilde{e}_1^I)\right)}. \quad (3.16)$$

The $h_m(f_1^J, e_1^I)$ in Equation 3.16 constitute a set of M different *feature functions*, each of which has an associated *scaling factor* λ_m . In this model, the inclusion of new models can be carried out by designing new feature functions. The structure of the model assures that we always stay in a correct mathematical formulation.

The denominator in Equation 3.16 is a normalization factor which is again independent of the translation e_1^I and can thus also be omitted. The resulting decision rule is:

$$f_1^J \rightarrow \hat{e}_1^I(f_1^J) = \operatorname{argmax}_{e_1^I} \{p(e_1^I | f_1^J)\} \quad (3.17)$$

$$= \operatorname{argmax}_{e_1^I} \left\{ \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, \tilde{e}_1^I)\right)} \right\} \quad (3.18)$$

$$= \operatorname{argmax}_{e_1^I} \left\{ \frac{\sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I)}{\sum_{\tilde{e}_1^I} \left(\sum_{m=1}^M \lambda_m h_m(f_1^J, \tilde{e}_1^I)\right)} \right\} \quad (3.19)$$

$$= \operatorname{argmax}_{e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I) \right\} \quad (3.20)$$

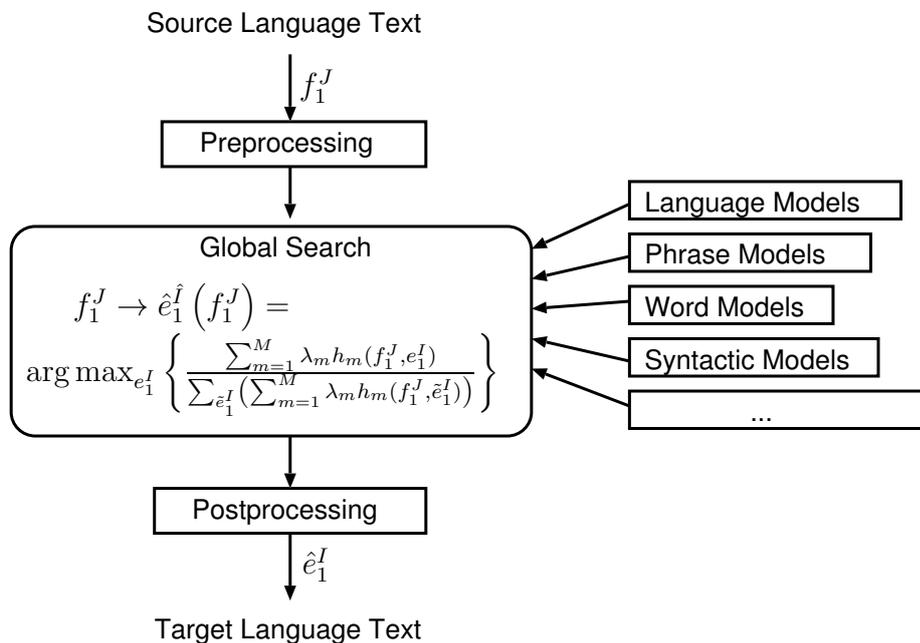


Figure 3.2: Illustration of the log-linear translation model. An arbitrary number of feature functions can be used in this approach.

The log-linear approach can be considered a generalization of Bayes' approach, i.e. we are able to include the language model and the translation model as before into the equation, but are also able to include a variety of other and possibly more complex models, and some of them will be discussed throughout this thesis. The architecture of a system using this approach is shown in Figure 3.2.

3.6 Optimization

Equation 3.20 introduces a scaling factor λ_m for each feature function. We want to assign those weights to the various models that result in the best translation performance on a held-out development set. Before we can review this optimization step, we need to define a quality measure for a hypothesis first. Human evaluation is too time-consuming and expensive to be used for system training, so we are looking for some automatic measures instead. These methods compare the hypothesis using the reference with some matching algorithm, and their correlation to actual human judgement is mostly considered to be satisfactory.

3.6.1 Error Measures and Scores

Possible error measures in SMT are the following:

WER The Word Error Rate (WER) is computed as the Levenshtein distance [Levenshtein 66] between a translation and a reference translation: the minimum number of required insertions, substitutions and deletions to match the two sentences will be divided by the reference length. Nowadays, the WER is hardly ever used as a main error measure since it is quite strict on the order of the words within a system.

PER The Position-Independent Word Error Rate (PER) is computed similar to the WER, but does not take the word position into account. This error measure thus is computed as the percentage of the words correctly translated, based on the reference.

TER The Translation Edit Rate (TER) as defined in [Snover & Dorr⁺ 06] is also derived from the WER, but allows for shifts of word blocks. It is a good trade-off between the order strictness of the WER and the order sloppiness of the PER and is one of the most common error measures used in the MT community.

BLEU The BiLingual Evaluation Understudy (BLEU) is computed as the n -gram precision of the hypothesis and its reference. It was described in [Papineni & Roukos⁺ 02]. Since a precision-based criterion tends to reward conservative output too much, a brevity penalty is also included so that short sentences still obtain bad scores. While not an error measure per se since good translations are rewarded with high scores, we use 1-BLEU internally but present BLEU in our experiments.

In this work, we are going to use the BLEU and the TER as quality measures, the current state-of-the-art in many international evaluation campaigns.

3.6.2 Minimum Error Rate Training

Now that we have defined quality measures for translation, we search for the best scaling factors that maximize the translation quality. This section will go a bit more into detail since many different techniques presented here are implemented in our open source toolkit Jane as presented in the next chapter. Much of this work was carried out as co-joint work with Markus Freitag.

We note that the error measures are neither linear functions nor differentiable with respect to λ , and we cannot use a gradient optimization method. However, there are some well-studied algorithms for gradient-free parameter optimization, e.g. the Downhill-Simplex method invented

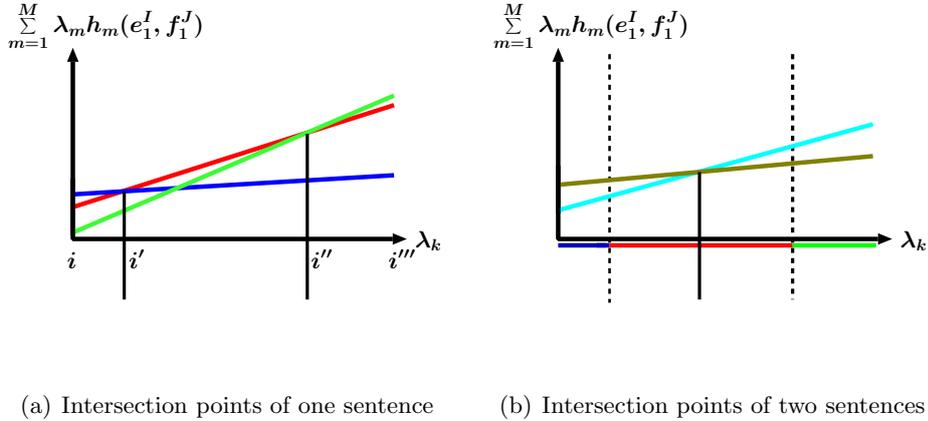


Figure 3.3: Calculation of intersection points in Och’s MERT algorithm. Figure (a) shows the 3-best list for one sentence and has three intersection points, of which only i' and i'' result in a change of the selected best sentence. Figure (b) shows the 2-best list of another sentence. Its intersection point will be added to the list of relevant intersection points.

by [Nelder & Mead 65] or Powell’s algorithm [Fletcher & Powell 63]. In our work, we use the method described in [Och 03]. We will denote it as Och’s method, although in the literature this method is usually called “Minimum Error Rate Training”. However, we find the term misleading since all the above optimization methods are computed with the goal of finding the minimal error.

Och’s method is derived from Powell’s algorithm. It works the output of a set of n -best hypotheses in a single decoder run and optimizes one scaling factor at a time, in a random order. The method exploits the fact that when changing only one scaling factor λ_k and keeping the others fixed, the translation score $f(\lambda) = \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)$ of one hypothesis is a linear function of one variable λ_k :

$$f(\lambda_k) = \lambda_k h_k(e_1^I, f_1^J) + \sum_{m=1, m \neq k}^M \lambda_m h_m(e_1^I, f_1^J) \quad (3.21)$$

We are only interested in the best translation within the n -best list, i.e. we only need to consider the intersection points of the upper envelope within the linear cost functions for each sentence (cf. Figure 3.3), which can be effectively computed by the sweep line algorithm [Bentley & Ottmann 79]. Computing the error measure for this limited set of intersection points, we select in each iteration the scaling factor which produces the translation with the lowest error.

The output of this form of optimization are only the optimized scaling

factors for the current n -best list. To obtain an optimization for a larger search space, we start different iterations of n -best list generation and optimization, and update the scaling factors in between. In each iteration we merge the generated n -best lists with the one of the previous iterations and optimize our scaling factors on this larger search space.

Jane: Open Source Hierarchical Decoder

In this chapter we will discuss the implementation of the hierarchical MT system, called Jane. It was developed from scratch at the RWTH Aachen University as part of this thesis and released officially in [Vilar & Stein⁺ 10a]. It includes all the features presented in this work, and is now freely available for non-commercial applications.

We will go through the main features of Jane, which include support for different search strategies, different language model formats, additional reordering models, extended lexicon models, different methods for minimum error rate training and distributed operation on a computer cluster. The syntactic methods, i.e. support for syntax-based enhancements to the hierarchical phrase-based machine translation paradigm as well as string-to-dependency translation, will only be briefly mentioned here, but are analyzed in more detail in Chapter 5. Results on four current MT tasks are reported, which show the system is able to obtain state-of-the-art performance on a variety of tasks.

This chapter is organized as follows: after an overview of the implementation in Section 4.1, Section 4.2 describes the core functionality of the toolkit, including parallelization capabilities. Section 4.3 introduces and describes advanced models for the translation system. Section 4.4 emphasizes the modular design of the code. In Section 4.5, Jane is compared to Joshua [Li & Callison-Burch⁺ 09], a decoder similar to our own, and the systems are measured in terms of translation performance and speed. The chapter is concluded in Section 4.6.

4.1 Implementation Overview

With Jane, we introduced a new open source toolkit for hierarchical phrase-based translation to the scientific community, free for non-commercial use. RWTH Aachen University has been developing this tool during the last few

years and it was used successfully in numerous MT evaluations, including NIST¹, WMT² and Quaero³ among others. It is developed in C++ with special attention to clean code, extensibility and efficiency.

Apart from the core algorithms, Jane implements many features presented in previous work developed both at RWTH Aachen University and other groups. In this chapter, we give an overview of the main features of the toolkit. References to relevant previous publications will be provided in the text as we discuss the different methods. We also discuss new extensions to the hierarchical model. Among them is an additional reordering model inspired by the distance-based reorderings widely used in phrase-based translation systems and another one comprises two extended lexicon models which further improve translation performance. We present experimental results for these methods on a variety of tasks.

4.2 Core Functionality

The preparation of a translation system normally involves four separate steps. First, the training corpus must be word-aligned (cf. Section 3.2.2). Well-established open-source programs exist for this (e.g. GIZA++ as described in [Och & Ney 03a]) and thus, no tools for this task are included in the Jane toolkit. In a second step, we have to extract the phrase pairs from the word-aligned parallel sentences, as discussed in Section 3.3.3. With the resulting phrase table, translations may be produced in a generation phase, as described briefly in Section 3.4. To obtain the best quality, however, some additional free parameters of the model have to be adjusted. As mentioned in Section 3.6.2, this is typically done by optimizing a quality measure on a held-out set of data, usually taken from the same domain as the sentences to be translated.

In this section, we will give an overview of how these steps are implemented in Jane. We will not go into full detail for previously published algorithms and methods, but refer to the given literature instead.

4.2.1 Extraction

The extraction of hierarchical phrases follows a two-step procedure. First, a set of initial phrases is extracted, as defined for the standard phrase-based approach (Eqn. 3.12). If a phrase is contained in a bigger phrase, the former is suppressed and a gap is created, producing a hierarchical phrase (Eqn. 3.15). This process is iterated until the desired maximum amount of gaps is produced. Probabilities for the phrases are computed as relative

¹<http://www.nist.gov/itl/iad/mig/mt.cfm>

²<http://www.statmt.org/>

³<http://www.quaero.org>

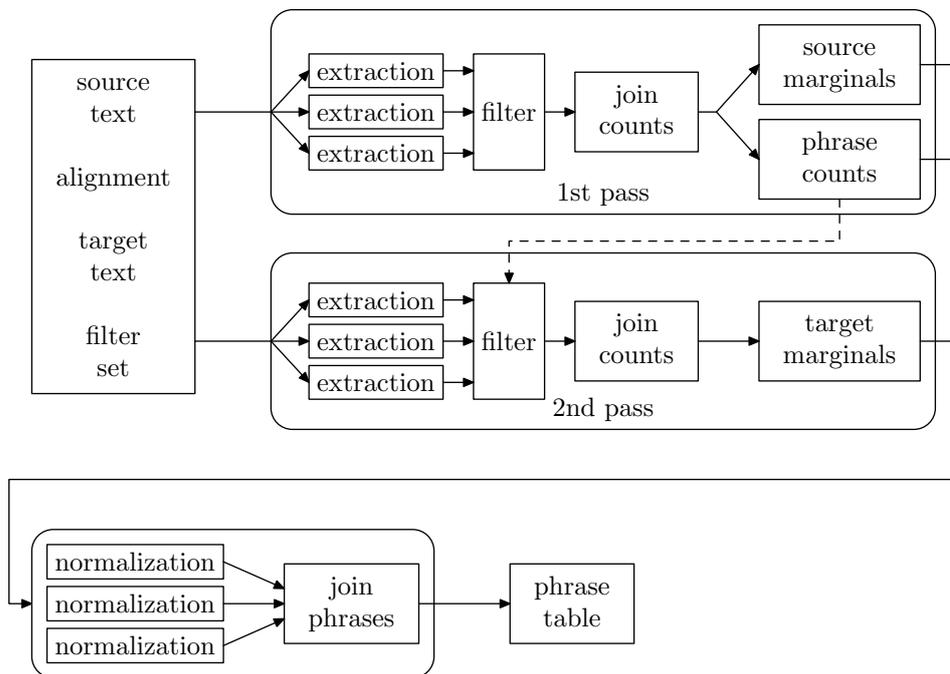


Figure 4.1: Workflow of the extraction procedure

frequencies. In order to reduce the size of the phrase table, Jane usually filters the extracted phrases to those needed for the translation of a given set of sentences.

Figure 4.1 shows a schematic representation of the extraction process. In a first pass we extract the bilingual phrases that are necessary for translating the given corpus. For normalization, two marginal counts are needed, since we would like to compute the relative frequencies for each side: the source marginals can be computed at the same time the phrases are extracted, whereas the target marginals are computed in a second pass, once the phrases have been extracted and the set of needed target marginals is known. By doing so, we can keep the size of the resulting files reasonable.

For parallelization the corpus is split into chunks, the granularity being user-controlled. Each of the necessary steps (count collection, marginal computation and count normalization) are then sent to separate nodes in a computer cluster. This operation, as well as the combination of the results, all happen automatically.

At extraction time, other information in addition to the relative frequencies can be produced. They may consist of features that can be computed at phrase-level, or they may consist of information that will be used in more complex models during generation. Currently implemented features include lexical smoothing based on IBM model 1, word and phrase penalties, binary

count features, phrase length ratios, Fisher’s significance and various binary features for sub-classes of hierarchical phrases. Additional information that can be appended to the phrase table entries include word alignment of the extracted phrases, dependency information, syntactic label information, number and position of unaligned words that were included in the phrase, internal gap size or part-of-speech information. Several of these entries are needed for the advanced models described in Section 4.3 and the syntactic models in Chapter 5. The implementation design allows easy integration of further custom information.

4.2.2 Generation

Once the extraction process is completed, we can start the actual translation procedure. Possible translations of a given source sentence are produced and scored using a probabilistic model, and the one with the highest probability will then be selected as the result (Eqn. 3.20). This search for the best-scoring translation proceeds in two steps. First, a monolingual parsing of the input sentence is carried out using the CYK+ algorithm [Chappelier & Rajman 98], a generalization of the CYK algorithm which relaxes the requirement for the grammar to be in Chomsky normal form. From the CYK+ chart we extract a hypergraph representing the parsing space (cf. Section 3.4).

In a second step the translations are generated, computing the language model scores in an integrated fashion. Both the cube pruning and cube growing algorithms [Huang & Chiang 07] are implemented. For the latter case, the extensions concerning the language model heuristics described in [Vilar & Ney 09] have also been included.

The translation process can be parallelized in a computer cluster. A series of jobs is started, the first one being the master job controlling the whole translation process. The sentences to be translated are submitted to the different computers in a dynamic fashion: when one computer has finished a translation, it notifies the master node and then receives the next sentence to translate. In order to better balance the load, longer sentences are the first ones sent to translate (while memorizing the target-language sentence order for reconstruction).

The same client-server infrastructure used for parallel translation may also be reused for interactive systems. Although no code in this direction is provided, one would only need to implement a corresponding front-end which communicates with the translation server, which may also be located on another machine.

Language Models

As described in Section 3.2.1, a language model is the a-priori probability distribution $Pr(e_1^I)$ over a string e_1^I , which in practice is typically restricted to n -grams rather than the whole sentence. Jane can handle four formats for n -gram language models: ARPA format, SRI binary format, randomized LMs and an in-house format. They differ mainly in their efficiency, both in terms of memory consumption and loading time.

The ARPA format for language models is supported using the SRI toolkit [Stolcke 02]. This format is the de-facto standard nowadays for storing language models. It is a plain-text based format and thus not especially optimized for machine operation. The SRI binary format, on the other hand, allows for a more efficient language model storage, which reduces loading time. We can take advantage of this characteristic in order to reduce memory consumption by reloading the LM for each sentence to translate, filtering out the n -grams that will not be needed for scoring the possible translations. The randomized LMs as described in [Talbot & Osborne 07] are a memory-efficient alternative, but come at the cost of a loss in accuracy. In particular the probability for unseen n -grams may be overestimated. We use the open source code made available by the authors of that paper. Jane’s in-house format is also memory-efficient, as it loads the required n -grams on-demand. However, this produces an increase in hard-disk accesses and thus the running time is increased. This format is implemented using the same internal prefix-tree implementation applied for phrase storage (see Section 4.4).

Several language models, also of mixed formats, can be used in parallel during translation. Their scores are combined in the log-linear framework.

4.2.3 Optimization Methods

Two main methods for minimum error rate training (cf. Section 3.6.2) are included in Jane. The first one is the procedure described in [Och 03], which has become a standard in the MT community. The second one is the MIRA algorithm, first applied for MT in [Chiang & Knight⁺ 09]. This algorithm is more adequate when the number of parameters to optimize is large. We use in-house implementations of both methods. Implemented is also the SPSA algorithm as described in [Spall & Member 92], first applied to MT in [Lambert & Banchs 06]. The optimization process also benefits from the parallelized translation operation (cf. Section 4.2.2). Additionally, for the minimum error rate training methods, random restarts may be performed on different computers in a parallel fashion (cf. also [Moore & Quirk 08]).

4.3 Advanced Models

In this section, we describe two additional models that go beyond a baseline hierarchical phrase-based system.

4.3.1 Additional Reordering Models

In the standard formulation of the hierarchical phrase-based translation model two additional rules are added:

$$\begin{aligned} S &\rightarrow \langle S^{\sim 0} X^{\sim 1}, S^{\sim 0} X^{\sim 1} \rangle \\ S &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \end{aligned} \tag{4.1}$$

This allows for a monotonic concatenation of phrases, very much in the way monotonic phrase-based translation is carried out. It is a well-known fact that for phrase-based translation, the use of additional reordering models is a key component, essential for achieving good translation quality (e.g. [Zens & Ney 06, Koehn & Arun⁺ 08]). In the hierarchical model, the reordering is already integrated in the translation formalism, but there are still cases where the required reorderings are not captured by the hierarchical phrases alone.

The flexibility of the grammar formalism allows us to add additional reordering models without the need to explicitly modify the code for supporting them. The most straightforward example would be to include the ITG-Reorderings [Wu 97] by adding the following rule:

$$S \rightarrow \langle S^{\sim 0} S^{\sim 1}, S^{\sim 1} S^{\sim 0} \rangle \tag{4.2}$$

We can also model other reordering constraints. As an example, phrase-level IBM reordering constraints with a window length of 1 can be included substituting the rules in Equation (4.1) with following rules:

$$\begin{aligned} S &\rightarrow \langle M^{\sim 0}, M^{\sim 0} \rangle \\ S &\rightarrow \langle M^{\sim 0} S^{\sim 1}, M^{\sim 0} S^{\sim 1} \rangle \\ S &\rightarrow \langle B^{\sim 0} M^{\sim 1}, M^{\sim 1} B^{\sim 0} \rangle \\ M &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \\ M &\rightarrow \langle M^{\sim 0} X^{\sim 1}, M^{\sim 0} X^{\sim 1} \rangle \\ B &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \\ B &\rightarrow \langle B^{\sim 0} X^{\sim 1}, B^{\sim 0} X^{\sim 1} \rangle \end{aligned} \tag{4.3}$$

In these rules we have added two additional non-terminals. The M non-terminal denotes a monotonic block and the B non-terminal a “back jump”. Actually both of them represent monotonic translations and the grammar could be simplified by using only one of them. Separating them allows for

Table 4.1: Results for the additional reorderings on the Europarl German-English data. BLEU and TER results are in percentage.

System	dev		test	
	BLEU	TER	BLEU	TER
Jane baseline	24.2	59.5	25.4	57.4
+ reordering	25.2	58.2	26.5	56.1

more flexibility, e.g. when restricting the jump width, where we only have to restrict the maximum span width of the non-terminal B . These rules can be generalized for other reordering constraints or window lengths. Additionally distance-based costs can be computed for these reorderings.

We tried this approach on the German-English language pair, for the Europarl task as defined in the Quaero project. The setting is very similar to the WMT evaluations. The results are shown in Table 4.1. As can be seen from these results, the additional reorderings obtain nearly 1% absolute improvement both in BLEU and TER scores.

4.3.2 Extended Lexicon Models

We enriched Jane with the ability to score hypotheses with discriminative and trigger-based lexicon models that use global source sentence context and are capable of predicting context-specific target words. This approach has recently been shown to improve the translation results of conventional phrase-based systems. In this section, we briefly review the basic aspects of these extended lexicon models. They are similar to [Mauser & Hasan⁺ 09], and we refer the reader there for a more detailed exposition on the training procedures and results in conventional phrase-based decoding.

Discriminative Word Lexicon

The first of the two lexicon models is denoted as a *discriminative word lexicon* (DWL) and acts as a statistical classifier that decides whether a word from the target vocabulary should be included in a translation hypothesis. For that purpose, it considers all the words from the source sentence, but does not take any position information into account, i.e. it operates on sets, not on sequences or even trees. The probability of a word being part of the target sentence, given a set of source words, is decomposed into binary features, one for each source vocabulary entry. These binary features are combined in a log-linear fashion with corresponding feature weights. The discriminative word lexicon is trained independently for each target word

using the L-BFGS [Byrd & Lu⁺ 95] algorithm. For regularization, Gaussian priors are utilized.

Let V_F be the source vocabulary and V_E be the target vocabulary. Then, we represent the source side as a bag of words by employing a count vector $\mathbf{F} = (\dots, F_f, \dots)$ of dimension $|V_F|$, and the target side as a set of words by employing a binary vector $\mathbf{E} = (\dots, E_e, \dots)$ of dimension $|V_E|$. Note that F_f is a count and E_e is a bit. The model estimates the probability $p(\mathbf{E}|\mathbf{F})$, i.e. that the target sentence consists of a set of target words given a bag of source words. For that purpose, individual models $p(E_e|\mathbf{F})$ are trained for each target word $e \in V_E$ (i.e. target word e should be included in the sentence, or not), which decomposes the problem into many separate two-class classification problems in the way shown in Equation (4.4).

$$p(\mathbf{E}|\mathbf{F}) = \prod_{e \in V_E} p(E_e|\mathbf{F}) \quad (4.4)$$

Each of the individual classifiers is modeled as a log-linear model:

$$p(E_e|\mathbf{F}) = \frac{e^{g(E_e, \mathbf{F})}}{\sum_{\tilde{E}_e \in \{0,1\}} e^{g(\tilde{E}_e, \mathbf{F})}} \quad (4.5)$$

with the function:

$$g(E_e, \mathbf{F}) = E_e \lambda_e + \sum_{f \in V_F} E_e F_f \lambda_{ef}, \quad (4.6)$$

where the λ_{ef} represent lexical weights and the λ_e are prior weights.

Triplet Lexicon

The second lexicon model we employ in Jane, the *triplet lexicon model*, is in many aspects related to IBM model 1, but extends it with an additional word in the conditioning part of the lexical probabilities. This introduces a better way of modelling long-range dependencies in the data. Like IBM model 1, the triplets are trained iteratively with the EM algorithm [Hasan & Ganitkevitch⁺ 08]. Jane implements the inverse triplet model $p(e|f, f')$.

The triplet lexicon model score $t(\cdot)$ of the application of a rule $X \rightarrow \langle \alpha, \beta \rangle$ where $\langle \alpha, \beta \rangle$ is a bilingual phrase pair that may contain symbols from the non-terminal set is computed as:

$$t(\alpha, \beta, f_0^J) = - \sum_e \log \left(\frac{2}{J \cdot (J+1)} \sum_j \sum_{j' > j} p(e|f_j, f_{j'}) \right) \quad (4.7)$$

with e ranging over all terminal symbols in the target part β of the rule. The second sum selects all words from the source sentence f_0^J (including the

empty word that is denoted as f_0 here). The third sum incorporates the rest of the source sentence to the right of the first triggering word. The order of the triggers is not relevant because per definition $p(e|f, f') = p(e|f', f)$, i.e. the model is symmetric. Non-terminals in β have to be skipped when the rule is scored.

In Jane, we also implemented scoring for a variant of the triplet lexicon model called the *path-constrained* (or *path-aligned*) triplet model. The characteristic of path-constrained triplets is that the first trigger f is restricted to the aligned target word e . The second trigger f' is allowed to move along the whole remaining source sentence. To be able to apply the model in search, Jane has to be run with a phrase table that contains word alignment for each phrase, with the exception of phrases which are composed purely of non-terminals (e.g. glue rules). Jane’s phrase extraction can optionally supply this information from the training data.

[Hasan & Ganitkevitch⁺ 08] and [Hasan & Ney 09] employ similar techniques and provide some more discussion on the path-aligned variant of the model and other possible restrictions. Table 4.2 shows the results for the French-English language pair of the Europarl task. On this task the extended lexicon models yield an improvement over the baseline system of 0.9% absolute (2.8% rel.) in BLEU and 0.9% absolute (1.8% rel.) in TER on the test set.

Table 4.2: Results for the extended lexicon models on the French-English task. BLEU and TER results are in percentage.

	dev		test	
	BLEU	TER	BLEU	TER
Baseline	30.0	52.6	31.1	50.0
DWL	30.4	52.2	31.4	49.6
Triplets	30.4	52.0	31.7	49.4
path-constrained Triplets	30.3	52.1	31.6	49.3
DWL + Triplets	30.7	52.0	32.0	49.1
DWL + path-constrained Triplets	30.8	51.7	31.6	49.3

We also show results on the Arabic-English NIST’08 task, using the NIST’06 set as development set. It has been reported in other work that the hierarchical system is not competitive with a phrase-based system for this language pair [Birch & Blunsom⁺ 09, Almaghout & Jiang⁺ 11]. We report the figures of our state-of-the-art phrase-based system as comparison

(denoted as PBT), in Table 4.3. The baseline Jane system is indeed 0.6% absolute worse (1.4% rel.) in BLEU and 1.0% absolute (2.0% rel.) worse in TER than the baseline PBT system. When we include the extended lexicon models we see that the difference in performance is reduced. For Jane the extended lexicon models give an improvement of up to 1.9% absolute (4.3% rel.) in BLEU and 1.7% absolute (3.2% rel.) in TER, respectively, bringing the system on par with the PBT system extended with the same lexicon models, and obtaining an even slightly better BLEU score.

Table 4.3: Results for the extended lexicon models for the Arabic-English task. BLEU and TER results are in percentage.

	test (MT'08)			
	Jane		PBT	
	BLEU	TER	BLEU	TER
Baseline	44.1	50.1	44.7	49.1
DWL	45.6	48.4	45.6	48.4
Triplets	45.3	48.8	44.9	49.0
path-constrained Triplets	44.9	49.3	45.3	48.7
DWL + Triplets	45.3	48.6	45.5	48.5
DWL + path-constrained Triplets	46.0	48.5	45.8	48.3

4.4 Extensibility

One of the goals when implementing the toolkit was to make it easy to extend it with new features. For this, an abstract class was created which we call a *secondary model*. New models need only to be derived from this class and implement the abstract methods for data reading and costs computation. This allows for an encapsulation of the computations, which can be activated and deactivated on demand. The models described in Sections 4.3.1 and 4.3.2, as well as some of the upcoming models in Chapter 5, are implemented in this way. We thus try to achieve loose coupling in the implementation, i.e. a system where each of its components has little or no knowledge of the definitions of other separate components.

In addition, a flexible prefix tree implementation with on-demand loading capabilities is included as part of the code. This class has been used for implementing the loading of phrases in the spirit of [Zens & Ney 07a] and the

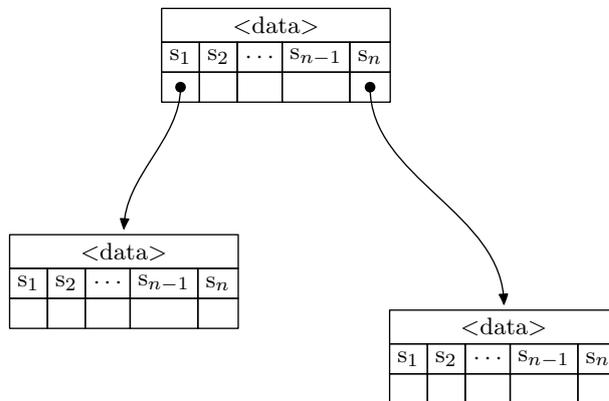


Figure 4.2: Implementation of a node in the prefix tree. The s_i denote the indexes of the successors of a node.

n -gram format described in Section 4.2.2, in addition to some intermediate steps in the phrase extraction process. The code may also be reused in other, independent projects.

The implementation of the data structure representing a node of the tree is depicted in Figure 4.2. The structure holds a data field, which in our case will be the set of translations for the given phrase. Conceptually, a list holds the labels of the arc connecting the successor nodes and a parallel list holds pointers to the corresponding nodes. In the actual implementation this may vary, e.g. by using vectors with implicit indexes.

If we want to store this structure in secondary memory for efficiency reasons, the pointers will be addresses on the disk. When loading the structure from disk we read it “as-is”, but marking the pointers as still being on secondary storage. If we need to follow a pointer, the corresponding node is loaded from disk and the pointer gets overwritten with an address in main memory.

4.5 Comparison with Joshua

As mentioned at the beginning of this chapter, Joshua is the most similar decoder to our own. It was developed in parallel at the Johns Hopkins University and it is currently used by a number of groups around the world.

Jane was started separately and independently. In their basic working mode, both systems implement parsing using a synchronous grammar and include language model information. Each of the projects then progressed independently, most of the features described in Section 4.3 being only available in Jane.

Efficiency is one of the points where we think Jane outperforms Joshua.

Table 4.4: Speed comparison Jane vs. Joshua, measured in translated words per second.

System	words/sec
Joshua	11.6
Jane cube prune	15.9
Jane cube grow	60.3

One of the reasons might well be the fact that it is written in C++ while Joshua is written in Java. In order to compare running times we automatically converted a grammar extracted by Jane to Joshua’s format and adapted the parameters accordingly. To the best of our knowledge we configured both decoders to perform the same task (cube pruning, 300-best generation, same pruning parameters). The results were equal except for some minor differences like e.g. the handling of the OOVs.

We tried this setup on the IWSLT’08 Arabic to English translation task. The speed results can be seen in Table 4.4. Jane operating with cube prune is nearly 50% faster than Joshua, at the same level of translation performance. If we switch to cube grow, the speed difference is even bigger, with a speedup of nearly 4 times. However this usually comes with a penalty in BLEU score, which based on our experience is commonly below 0.5% BLEU for most tasks. This increased speed can be specially interesting for applications like interactive machine translation or online translation services (cf. [Zens & Ney 07b]), where the response time is critical and sometimes even more important than a small and often hardly noticeable loss in translation quality.

Another important point concerning efficiency is the startup time. Thanks to the binary format described in Section 4.4, there is virtually no delay in the loading of the phrase table in Jane, and only a small delay for some of the supported language models.

For comparison of translation results, we refer to the results of the WMT 2010 evaluation shown in Figure 4.5. John Hopkins University participated in this evaluation using Joshua, the system was trained by its original authors [Schwartz 10] and thus can be considered to be fully optimized. RWTH also participated using Jane among other systems. A detailed description of RWTH’s submission can be found in [Heger & Wuebker⁺ 10]. The scores are computed using the official euromatrix web interface for machine translation evaluation.⁴

As can be seen the performance of Jane and Joshua is similar, but Jane

⁴<http://matrix.statmt.org/>

Table 4.5: Results for Jane and Joshua in the WMT 2010 evaluation campaign.

	Jane		Joshua	
	BLEU	TER	BLEU	TER
German-English	21.8	69.5	19.5	66.0
English-German	15.7	74.8	14.6	73.8
French-English	26.6	61.7	26.4	61.4
English-French	25.9	63.2	22.8	68.1

generally achieves better results in BLEU, while Joshua has an advantage in terms of TER. Having different systems is always enriching, and particularly as system combination shows great improvements in translation quality, having several alternative systems can only be considered a positive situation. Also note that some of the differences in the evaluation measures can be due to the optimization procedure (confer e.g. [He & Way 09]), which is hard to trace for different systems.

4.6 Conclusion

Jane is a state-of-the-art hierarchical toolkit made available to the scientific community. The system in its current state is stable and efficient enough to handle even large-scale tasks such as the WMT and NIST evaluations, while producing highly competitive results. The system implements the standard hierarchical phrase-based translation approach and different extensions that further enhance the performance of the system. Some of them, like additional reordering and lexicon models are exclusive to Jane.

In general, however, we feel that the hierarchical phrase-based translation approach still shares some shortcomings concerning lexical selection with conventional phrase-based translation. Bilingual lexical context beyond the phrase boundaries is barely taken into account by the base model. In particular, if only one generic non-terminal is used, the selection of a sub-phrase that fills the gap of a hierarchical phrase is not affected by the words composing the phrase it is embedded in, with the exception of the language model score.

The extended lexicon models analyzed in Section 4.3.2 already try to address this issue. One can consider that they complement the efforts that are being made on a deep structural level within the hierarchical approach. Though they are trained on surface forms only, without any syntactic information, they still operate at a scope that exceeds the capability of common feature sets of standard hierarchical phrase-based SMT systems.

In the next chapter, we address the shortcoming of the hierarchical approach with syntactically motivated models.

Soft Syntactic Features

Hierarchical phrase-based translation as introduced in Section 3.4 and analyzed more thoroughly in the last chapter, has proven to be one of the most successful approaches for SMT. The approach can be considered as a formal syntactic model, since the underlying structure is a grammar lacking linguistic knowledge. Given the increasing availability of linguistic parsers for many languages, hybrid approaches which incorporate deep syntactic knowledge often improve the translation quality. The goal is to enforce a more fluent grammatical structure on the output hypotheses. Various groups report improvement over their baseline systems with different approaches, but it is not clear whether the benefits of the different methods are complementary or if they rather address the same issues.

This chapter is organized as follows. After a general introduction to syntactic parsing in Section 5.1, we will present and review three syntactically motivated enhancements to the hierarchical translation system: in Section 5.2, we start with a relatively simple model that measures how close a translation phrase is to the yield of a given parse tree. Section 5.3 then reviews soft syntactic labels, where the phrases are marked with syntactic labels in an additional feature, trying to improve over the generic and thus somewhat arbitrary non-terminal used in the hypergraph derivation. In Section 5.4, we will present our implementation decisions on a dependency-based language model that is able to score words that span a wider range. We apply the models on various language pairs. The results are presented and analyzed in Section 5.5.

The experiments in this section were supported by Stephan Peitz and Jan-Thorsten Peter.

5.1 Syntactic Parsing

When representing the structure of a sentence from a linguistic viewpoint, there is the possibility to employ a phrase structure parse which represents some nesting of multi-word constituents such as noun phrases which may include the noun, its adjective and its article. Another way of representing the sentence is with a dependency parse which represents dependencies between individual words, which can be either generic, or labelled when the relation between a head and its dependent can be identified more precisely, e.g. for modifiers or a subject relation.

We parse the English target sentences with the Stanford parser,¹ which is able to produce phrase structure parses [Klein & Manning 03] like in Figure 5.1(a). The Stanford parser also offers dependency structures as well [de Marneffe & MacCartney⁺ 06], like in Figure 5.2(a).

For French as the target language, we extended Jane to include French dependencies via the freely available Bonsai parser [Candito & Crabb⁺ 10], which makes use of the Berkeley parser as described in [Petrov & Barrett⁺ 06].

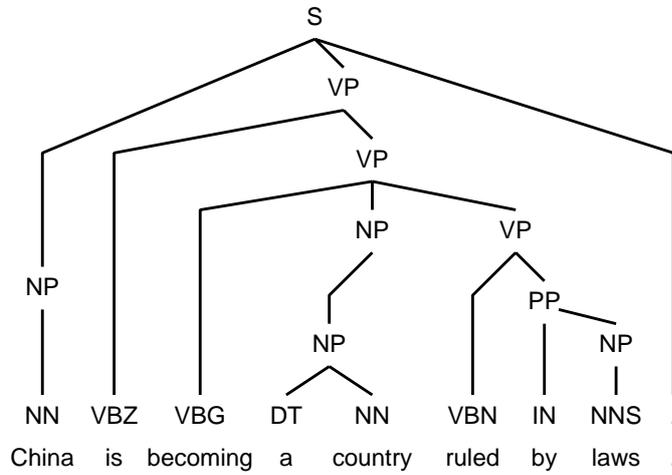
5.2 Parse Matching

The first model that we employ is also the simplest one. Given a monolingual sentence, be it in the source or the target language, and the associated parse tree, we say that a lexical phrase extracted from this sentence is syntactically valid if it corresponds to the yield of one of the nodes in the syntax tree.

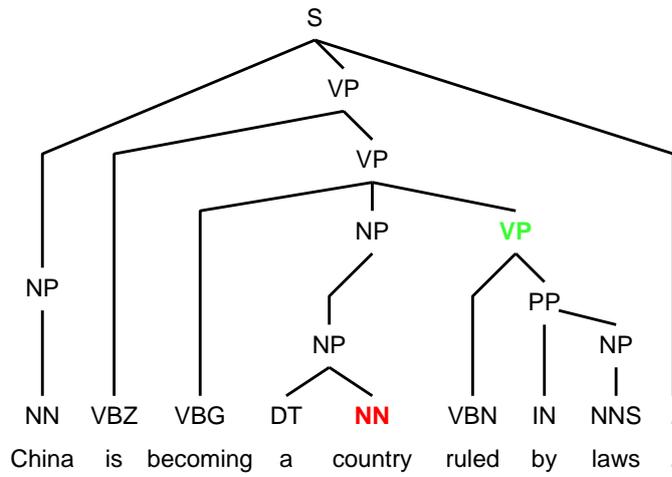
With this model, we hope that we can guide the decoder to prefer phrases that are syntactically sound rather than using arbitrary word combinations that spread over the boundaries of syntactic constructs. Two features are derived from this procedure. The first one measures the relative frequency with which a given phrase does not exactly match the yield of any node. This feature is straightforward to compute for the initial phrases. We extend this concept to hierarchical phrases by considering them as valid if the originating initial phrase was syntactically valid and every phrase which was suppressed in order to generate the gaps was also syntactically valid.

In the second feature, we soften up this rather hard decision. We might want to penalize those phrases that only miss a valid node by one word less than those that have a bigger mismatch with the parse tree. For example, in Figure 5.1(b), “country ruled” should be considered worse on a phrasal level than “country ruled by laws”. The second feature thus measures the relative distance to the next valid node, i.e. the average number of words that have to be added or deleted to match a syntactic node, divided by the phrase length. Hierarchical phrases are treated in a similar way as above. This

¹<http://nlp.stanford.edu/software/lex-parser.shtml>

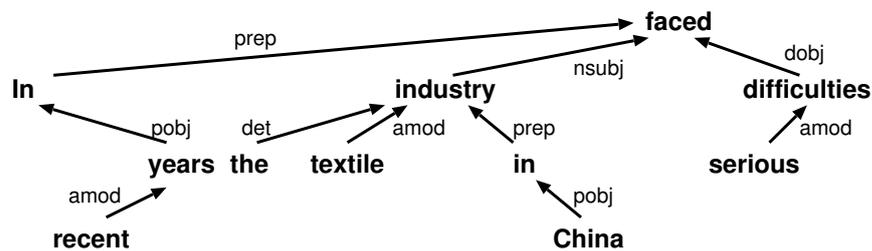


(a) Phrase structure parse. Labels: determiner (DT), preposition or subordinating conjunction (IN), single noun (NN), noun plural (NNS), noun phrase (NP), prepositional phrase (PP), simple declarative clause (S), verb, gerund or present participle (VBG), verb, past participle (VBN), verb, 3rd person singular present (VBZ), verb phrase (VP).



(b) Visualization for the feature criteria derived in Section 5.2 and Section 5.3. "ruled by laws" matches the yield of the noun phrase (NP) label and would be considered valid, whereas "country ruled" would be considered invalid and matched to the single noun (NN) label with a distance measure of .5.

Figure 5.1: Stanford phrase structure parse for the sentence "China is becoming a country ruled by laws."



(a) Dependency parse. Labels: adjectival modifier (amod), determiner (det), direct object (dobj), nominal subject (nsubj), object of a preposition (pobj), prepositional modifier (prep).



(b) For the word “faced”, three probabilities will be computed: $p_{\text{left}}(\text{faced}|\text{industry}, \text{in})$, $p_{\text{head}}(\text{faced})$, $p_{\text{right}}(\text{faced}|\text{difficulties})$.

Figure 5.2: Stanford dependency parse for the sentence “In recent years, the textile industry in China faced serious difficulties.” and an example for some of the resulting dependency LM probabilities.

approach is similar to the “binary” and “relative” soft syntactic features as we described in [Vilar & Stein⁺ 08a].

See Figure 5.3 for an overview of the average distance to the next valid phrase node, for the different translation directions. Chinese–English has 27% valid rules in its phrase table, and already 47% of rules that have a word distance of one or less. For German–French, the percentages follow a similar trend but are generally 7-10% lower than that, i.e. fewer rules are marked valid. Among the selected language pairs, the Arabic–English setting has the worst matching rules: only 7% of the phrases match the yield of a node, with 20% having a word distance of one or less.

5.3 Soft Syntactic Labels

Another possibility to extend the hierarchical model and include syntax information is to extend the set of non-terminals in the hierarchical model from the original set of generic symbols to a more rich, syntax-oriented set. With this, we hope to improve the syntactic structure of the output sentence. For example, there may be rules which ensure that there is a verb in the translation of every source verb phrase.

However, augmenting the set of non-terminals also restricts the parsing space and thus we alter the set of possible translations. Furthermore, it

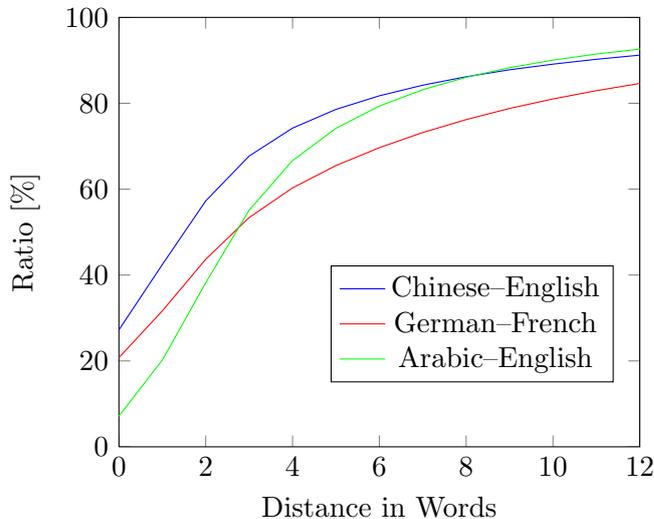


Figure 5.3: All corpora: percentage of the distinct phrase table entries having a phrase match word distance of value x or lower.

can happen that no parse can be found for some input sentences. To address this issue, our extraction is extended in a similar way to the work of [Venugopal & Zollmann⁺ 09]. In this model, the original generic non-terminals A and B are not substituted, but rather the new non-terminals are appended as additional information to the phrases and a new feature is computed based on them. In this way the original parsing and translation spaces are left unchanged. In contrast to the above work, where the authors expand the set of linguistic non-terminals to include a large set of new symbols, we restrict ourselves to the non-terminals that are found in the syntax tree.

Each initial phrase is marked with the non-terminal symbol of the closest matching node as described in Section 5.2. When producing hierarchical rules, the gaps become labelled with the non-terminal symbols of the corresponding phrases instead of the original generic non-terminals A and B . It is important to point out that the syntax information is extracted from the target side only, but the substitution of the corresponding non-terminal symbol is carried out both on the source and target sides (with the same non-terminal on both sides).

For every rule in the grammar, we store information about the possible non-terminals that can be substituted in place of the generic non-terminals A and B , together with a probability for each combination of non-terminal symbols. More formally, let S be the set of possible syntax non-terminals. Given a rule r with n gaps, we define a probability distribution $p(s|r)$ over $S^n + 1$, where s denotes a possible combination of syntax non-terminal sym-

bols to be substituted in the rule, including the left-hand-side.

For each derivation d we compute two additional quantities. The first one is denoted by $p_h(Y|d)$ (h for “head”) and reflects the probability that the derivation d under consideration of the additional non-terminal symbols has $Y \in S$ as its starting symbol. This quantity is needed for computing the probability $p_{\text{syn}}(d)$ that the derivation conforms to the extended set of non-terminals.

For the exact definition of these two quantities we separate the case where the top rule of derivation d is an initial phrase (in which case the derivation consists only of one rule application) and the general case where the top rule is a hierarchical one. If the top rule r of d corresponds to an initial phrase, the probability distribution for the non-terminals for d will be equal to the distribution of rule r , i.e. $p_h(s|d) = p(s|r), \forall s \in S$. Given that only one rule has been applied, the derivation fully conforms to the extended set of non-terminals, so in this case $p_{\text{syn}}(d) = 1$.

For the general case of hierarchical rules, let d be a general derivation, let r be the top rule and let d_1, \dots, d_n be the sub-derivations associated with the application of rule r in derivation d . For determining whether the derivation is consistent with the extended set of non-terminals we have to consider every possible substitution of non-terminals in rule r and check the probability of the n sub-derivations to have the corresponding non-terminals.

More formally:

$$p_{\text{syn}}(d) = \sum_{s \in S^{n+1}} \left(p(s|r) \cdot \prod_{k=2}^{n+1} p_h(s[k]|d_{k-1}) \right), \quad (5.1)$$

where the notation $[\cdot]$ denotes addressing the elements of a vector. The index shifting in the product in Equation 5.1 is due to the fact that the first element in the vector of non-terminal substitutions is the left-hand-side of the rule, and this has to be taken into account when multiplying with the probabilities of the sub-derivations. Note also that although the sum is unrestricted, most of the summands will be left out due to a zero probability in the term $p(s|r)$.

The probability p_h is computed in a similar way, but the summation index is restricted only to those vectors of non-terminal substitutions where the left-hand side is the one for which we want to compute the probability. More formally:

$$p_h(Y|d) = \sum_{s \in S^{n+1}: s[1]=Y} \left(p(s|r) \cdot \prod_{k=2}^{n+1} p_h(s[k]|d_{k-1}) \right). \quad (5.2)$$

5.4 Soft String-To-Dependency

Given a dependency tree of the target language, we are able to introduce language models that span over longer distances than shallow n -gram language models (see Figure 5.2(b)).

Rather than parsing the structures during decoding, we apply the Stanford parser already on the training material. Here, the sentences are generally well-formed and produce no additional parser noise (on-top of errors introduced by a potentially erroneous parser), as opposed to an n -best list of the hypotheses. [Shen & Xu⁺ 08] use only phrases that meet certain restrictions. The first possibility is what the authors called a *fixed* dependency structure. With the exception of one word within this phrase, called the *head*, no outside word may have a dependency within this phrase. Furthermore, all inner words may only depend on each other or on the head. For a second structure, called a *floating* dependency structure, the head dependency word may also exist outside the phrase. More formally, let dep_j denote the dependency of a word j on another word. A dependency structure $dep_{i\dots j}$ is called *fixed* on head h iff

- $dep_h \notin [i, j]$
- $\forall k \in [i, j] \wedge k \neq h, dep_k \in [i, j]$
- $\forall k \notin [i, j], dep_k = h \vee dep_k \notin [i, j]$

and *floating* with children C for a non-empty set $C \subseteq \{i, \dots, j\}$ iff

- $\exists h \notin [i, j], \text{ s.t. } \forall k \in C, dep_k = h$
- $\forall k \in [i, j] \wedge k \notin C, dep_k \in [i, j]$
- $\forall k \notin [i, j], dep_k \notin [i, j]$.

See Table 5.1 for statistics of the dependency structures based on the phrase table size. Phrases that are marked as *fixed on head* only form a very small section of the corpus, around 2.5% for Chinese–English, 2.2% for German–French and a mere 1.6% for Arabic–English. The difference in the language pairs extends to the phrases that are *floating with children*: almost half of the phrases, 41.1% (Chinese–English), would still be considered valid by the definition in [Shen & Xu⁺ 08]. This is in contrast to their findings that valid phrases make up $\approx 20\%$ of the phrase table, a proportion that we only witness for Arabic–English (19.4%) and approximatively for German–French (27.1%). Possible reasons for this are the different parsers employed (no exact details are given in the original paper) or handling of punctuation marks (which are not parsed by the Stanford parser). Another reason might be that we filter the phrase table according to the test sets. This might result in fewer “random” phrases being extracted from the training set.

Table 5.1: All corpora: statistics for the dependency structures labelled “fixed on head” and “floating with children” (Section 5.4), based on the overall phrase table size. Note that our phrase tables are filtered to contain phrases from the translation sets only.

	total	# phrases	
		fixed on head	floating
Chinese–English [Shen & Xu ⁺ 08]	140 M	27 M (19.2%)	
Chinese–English	43,367,641	1,100,432 (2.5%)	17,827,382 (41.1%)
Arabic–English	67,023,448	1,092,339 (1.6%)	13,020,791 (19.4%)
German–French	34,317,691	781,925 (2.2%)	9,310,446 (27.1%)

In our phrase table, we mark all phrases deemed valid dependency structures with a binary feature, but again do not limit the total phrase translation entry table. Additionally, we store the dependency information in our phrase table, and further memorize for all hierarchical phrases whether the gaps were dependent on the left or on the right side.

The approach in [Shen & Xu⁺ 08] relies on reconstructing the dependency tree of a hypothesis at decoding time by a bottom-up approach, and for their algorithm they rely on valid phrases only, and discard all others. The authors note in their paper that simple filtering does not yield improvements, but rather causes the results to deteriorate. It is only by employing the language model on the resulting dependency structure that they are able to improve significantly over the baseline.

For this reason, and also to be in line with our previous soft features, we extended the merging step to be able to work with “invalid” phrases as well. For this, we perform a normal bottom-up merging of the dependencies (Algorithm 1), where we recursively process the derivations in the hypernodes. Whenever we are not in a leaf, we might encounter sub-structures that are not yet expanded and might shift the dependency positions on the right hand-side. Thus, we keep track of the resulting offset whenever the sub-structure is larger than 1, and compute their offset later in Algorithm 2. The actual error handling happens in Algorithm 3. Here, we check and penalize whenever the child dependency structure filling the gap in a parent hierarchical structure is pointing in the wrong direction, i.e. the child is pointing to the left when the parent rule is expecting the gap to point to the right, or vice versa (see Figure 5.4). Note that this can only happen when the child dependency is pointing outside of its phrase (denoted by `isPointer` in the algorithm).

We include three features in our log-linear model: merging errors to the left, merging errors to the right and the accumulated ratio of non-valid dependency structures used.

Algorithm 1: mergeDependency

```

input : DependencyTreeNode node
output: Merged Tree returnList
children = node.getChildren;
for each dependency in children do
  if dependency.isBranch then
    | dependency += computeOffset (dependency);
    | returnList.pushBack (dependency);
  else
    | mergedList = mergeDependency (dependency);
    | offsetList.pushBack (returnList.size (), mergedList.size () -1);
    | adjustPointers (mergedList, returnList, dependency);
return returnList;

```

Algorithm 2: computeOffset

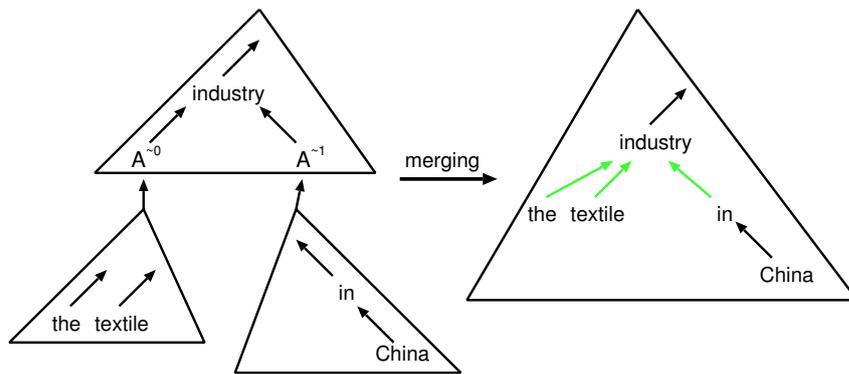
```

input: Dependency pointer dependency, global list offsetList
if dependency.isNumber() then
  | for offset in offsetList do
  | | if offset.position < dependency then
  | | | dependency += offset.size;
return dependency;

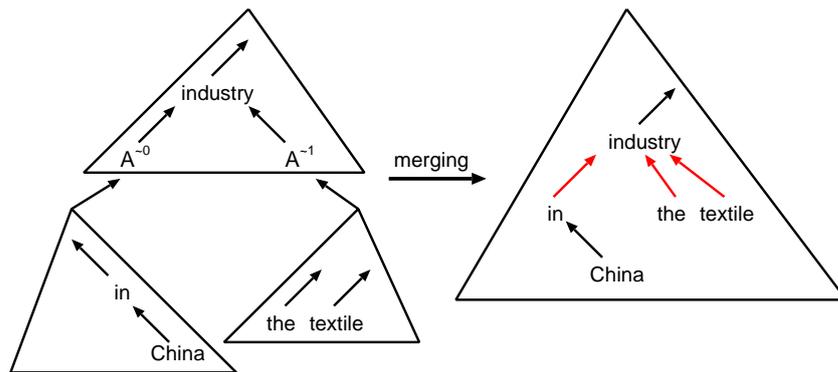
```

In a subsequent step, we further compute the probabilities of the dependency structures using two language models: one for left-side dependencies, and one for right-side dependencies. For head structures, we also compute their probabilities by exploiting a simple unigram language model. In early experiments, we noticed that badly reconstructed dependency trees have fewer probabilities to compute and thus tend to score higher than better structured trees in other sentences. We decided to include a language count feature that is incremented each time we compute a dependency language model score, similar to the word penalty used for the normal language model.

Note that the language model step was only implemented as a rescoring approach but has been extended into an online step in [Peter & Huck⁺ 11].



(a) Merging without errors.



(b) Merging with three errors

Figure 5.4: Merging errors in dependency tree reconstruction. In (a), the inserted dependency structures fit into the higher structure, and no merging error occurs. In (b), neither derivation has the same direction as the higher structure, so that three dependency directions need to be adopted.

Algorithm 3: adjustPointers

```
input: merged list mergedList, old list returnList, dependency of
parent structure parentDependency
for each dependency in mergedList do
  if dependency.isNumber() then
    // i.e. pointing to a concrete position
    adjustedList.pushBack ( dependency + returnList.size());
  else if dependency.isPointer() then
    // i.e. a head or pointing to the left/right
    if parentDependency.isNumber() then
      check whether direction matches dependency;
      newDependency = computeOffset ( parentDependency);
      adjustedList.pushBack ( newDependency);
      if parentDependency points to unexpanded sub-structure
      then
        remember to correct pointer later;
    else if parentDependency.isPointer() then
      check for left/right errors;
      adjustedList.pushBack ( parentDependency);
  else
    // invalid position, e.g. unparsed punctuation marks
    adjustedList.pushBack ( dependency);
return adjustedList;
```

5.5 Experiments

In this section, we present the results for the individual models on the NIST Chinese–English task, and on the QUAERO German–French task.

For Chinese–English, the NIST 2006 was used as the development set for minimum error training on BLEU in all the experiments. We presented results in [Stein & Peitz⁺ 10], we worked with a smaller 6-gram LM that had a perplexity of 113.7 on the NIST ’06 test set. NIST ’08 was used as our test set. In more recent experiments, we employed a larger 6-gram language model with a perplexity of 106.56 on the ’06 test set. In this setting, results are checked on the combined ’02+’04 test set, the ’05 test set and the ’08 set. For a detailed corpus overview, see Table B.1 on page 102.

In Table 5.2, we present the results for the three methods. For the experiments in [Stein & Peitz⁺ 10], all but the parse match method yield significant improvements over the baseline. With the newer LM, all methods yield statistically significant improvement over the baseline in almost all test sets. The highest gain in terms of BLEU and TER is usually achieved by the syntactic label approach. In Table 5.3, the dependency model is examined more closely (using the newer conventional LM, and two dependency trigram LMs derived from the training data). When only applying the dependency language model during rescoring, statistically significant improvement is mostly seen in terms of TER, but the BLEU improvement is not statistically significant, and the score even deteriorates slightly for NIST ’08. If the system is optimized with the additional merging error features of the tree reconstruction, it seems that the dependency trees are more well-formed; the dependency LM rescoring performs better in all tasks.

In Table 5.4, where we combine all methods, the influence of the models seems to be complementary, as the performance usually increases with each individual model. For the newer LM setting, however, the combination of all models still falls short in comparison to the single syntactic labels system, an effect that is not present for the [Stein & Peitz⁺ 10] experiment.

See Table 5.5 for examples where the translation quality is improved in the NIST ’08 task (e.g. “I is not an easy miffed” becomes “I am a person who is not easy miffed”). Note that for these particular examples, the improvement in BLEU or TER is not obvious, whereas the fluency of the translation is clearly improved.

For German–French, the development set of 2010 was used, and it was tested on the test ’10 and the eval ’10 set. For a detailed corpus overview, see Table B.3 on page 103. See Table 5.6 for the results for the different methods on this task, and Table 5.7 for a more detailed dependency feature analysis. In both tables, the improvements are only slight and not statistically significant. By combining all methods (Table 5.9), we are able to obtain statistically significant better scores, which are highly competitive (see also Table 5.10 for a system comparison within the QUAERO project).

Table 5.2: Results for the NIST Chinese–English task with all syntactic models. Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$.

Setting	nist06 (dev)		nist02/04		nist05		nist08	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline [Stein & Peitz ⁺ 10]	31.4	63.2					24.0	68.4
90% significance	± 0.7	± 0.6					± 0.7	± 0.6
parse match	31.4	63.1					24.4	67.9
syntactic labels	32.2	62.1					25.0	67.2
dependency with rescoring	32.2	61.9					24.6	66.7
baseline, better LM	32.3	62.6	33.5	62.1	31.3	63.8	25.3	67.5
90% significance	± 0.7	± 0.6	± 0.5	± 0.5	± 0.9	± 0.9	± 0.7	± 0.6
parse match	33.1	62.2	34.1	61.5	32.1	62.7	25.7	67.3
syntactic labels	33.4	61.2	35.7	59.2	33.6	60.4	25.7	66.5
dependency with rescoring	33.2	61.5	34.4	61.0	32.5	62.0	25.9	66.3

Table 5.3: Results for the NIST Chinese-English task with the soft string-to-dependency model. Rescoring and Oracle Score on 1 k n -best lists. Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$.

Setting	nist06 (dev)		nist02/04		nist05		nist08	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline (base)	32.3	62.6	33.5	62.1	31.3	63.8	25.3	67.5
+ LM rescoring	32.9	61.8	34.1	61.2	32.0	62.2	25.1	67.0
dependency (dep)	32.5	62.2	33.3	62.1	31.4	63.8	26.1	66.7
+ LM rescoring	33.2	61.5	34.4	61.0	32.5	62.0	25.9	66.3
base oracle	47.1	47.4	48.2	48.1	45.9	48.9	38.5	52.2
dep oracle	47.4	47.3	48.1	48.2	45.7	49.3	39.4	51.7

Table 5.4: Results for the NIST Chinese–English task with syntactic model combination.

Setting	nist06 (dev)		nist02/04		nist05		nist08	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline [Stein & Peitz ⁺ 10]	31.4	63.2					24.0	68.4
90% significance	± 0.7	± 0.6					± 0.7	± 0.6
+ parse match	31.4	63.1					24.4	67.9
+ syntactic labels	32.4	62.3					25.3	67.3
+ dependency with rescoring	32.9	61.0					25.1	66.4
baseline, better LM	32.3	62.6	33.5	62.1	31.3	63.8	25.3	67.5
90% significance	± 0.7	± 0.6	± 0.5	± 0.5	± 0.9	± 0.9	± 0.7	± 0.6
+ parse match	33.1	62.2	34.1	61.5	32.1	62.7	25.7	67.3
+ syntactic labels	33.1	61.7	34.6	61.2	32.9	61.5	25.7	66.7
+ dependency with rescoring	33.4	61.9	34.8	60.2	33.3	60.8	25.5	66.7

5.6 Conclusion

In general, the syntactic methods presented in this chapter all lead to modest to high improvements over their respective baseline; none worsens the error measures. It can be argued that this is due to the fact that all the features are soft, so that the decoder can always fall back to the baseline if the syntactic models do not offer much help for the particular sentence. The highest gain can be seen for the language pair Chinese–English. Even with a stronger baseline system due to a better conventional language model, the gain in performance is similar. The parse match model is the typically the weakest of the three, but never hurts in performance, is easy to implement and hardly takes up additional computation time during decoding. The gain by the syntactic label model is higher, at the cost of higher memory requirements (≈ 2.5 GB). The gains from the dependency model seems two-fold: some improvement can be seen due to the penalization of left/right merging errors, while others are due to the language model rescoring.

For German–French there is also some gain over the baseline, albeit not as pronounced as for Chinese–English. The tendencies appear to be similar, however. A possible reason is that the grammatical structure of these European languages have more similarities and thus do not profit as much from the syntactic enhancements.

Since the methods presented have been designed to produce soft features, we are able to run them in parallel on the same data. In the full system,

Table 5.5: NIST Chinese–English: examples for translations with a higher syntactic soundness when relying on soft syntactic features.

(a) Example 1	
source	当然,但愿这一切担心都是多余的
glosses	of course hopefully this all worries are redundant
reference	Of course, I hope that all of these worries are needless.
baseline	Of course, I hope that all these are worried that is.
syntactic models	Of course, I hope that all this worry is superfluous.
(b) Example 2	
source	我想我是一个不容易生气的人, 起码我自己的感觉是这样的。
glosses	I think I am a not easy get angry people at least I own feeling is this
reference	I think I'm someone who doesn't easily lose his temper. That's my own feeling, at least.
baseline	I would like to the people I is not an easy miffed, at least , I feel this is the case.
syntactic models	I think I am a person who is not easy miffed, at least I have the feeling that's the way it is.

Table 5.6: Results for the QUAERO German–French task with all syntactic models.

Setting	dev		test		eval '10	
	BLEU	TER	BLEU	TER	BLEU	TER
baseline	20.8	67.6	21.0	67.3	36.2	53.1
90% significance	±0.5	±0.6	±0.5	±0.7	±0.8	±0.7
parse match	20.7	67.1	21.1	66.7	36.5	51.9
syntactic labels	21.1	66.7	21.7	66.0	36.8	52.0
dependency with rescoring	21.0	67.0	21.3	66.6	36.3	52.6

Table 5.7: Results for the QUAERO German–French task with the soft string-to-dependency model. Rescoring and Oracle Score on 1 k n -best lists.

Setting	dev		test		eval	
	BLEU	TER	BLEU	TER	BLEU	TER
baseline	20.8	67.6	21.0	67.3	36.2	53.1
baseline with rescoring	21.1	67.0	21.2	66.8	36.2	52.9
dependency	20.9	66.9	21.3	66.6	36.3	52.6
dependency with rescoring	21.0	67.0	21.3	66.6	36.3	52.6
baseline oracle	31.6	53.9	32.4	53.0	50.0	38.6
dependency oracle	31.8	53.2	32.6	52.5	50.3	38.2

Table 5.8: QUAERO German–French: examples for translations with a higher syntactic soundness when relying on soft syntactic features.

(a) Example 1

source	Aber Inflation verursachen sie nicht, sofern sie nicht zu einem Überhang der Nachfrage nach Waren und Arbeitskräften führen.
reference	Mais ils ne causent pas d’inflation, tant qu’ils ne mènent pas à un excédent de la demande pour les biens et la main-d’œuvre.
baseline	L’inflation mais ils ne sont pas, pour autant qu’elle ne devienne pas un dépassement de la demande de biens et de main-d’œuvre.
syntactic models	Mais ils ne provoquent pas l’inflation, pour autant qu’elle ne devienne pas un dépassement de la demande de biens et de main-d’œuvre.

(b) Example 2

source	Diese schon lange vergessenen Kämpfe scheinen plötzlich wieder sehr aktuell zu sein .
reference	Ces batailles oubliées depuis longtemps semblent soudainement très présentes à nouveau .
baseline	Ces combats sont de nouveau très d’ actualité semblent oubliées depuis longtemps .
syntactic models	Ces combats depuis longtemps oubliées semblent soudain très actuel .

Table 5.9: Results for the QUAERO German–French task with all syntactic models.

Setting	dev		test		eval '10	
	BLEU	TER	BLEU	TER	BLEU	TER
baseline	20.8	67.6	21.0	67.3	36.2	53.1
+ parse match	20.7	67.1	21.1	66.7	36.5	51.9
+ syntactic labels	20.9	66.6	21.5	66.2	36.8	51.8
+ dependency with rescoring	21.2	66.8	21.6	66.2	37.0	52.0

Table 5.10: QUAERO German–French: Comparison of the translation performance within the QUAERO project, for the evaluation in 2010.

Setting	eval '10	
	BLEU	TER
LIMSI (Paris) submission	33.2	54.7
KIT (Karlsruhe) submission	36.0	52.3
RWTH best single system 2010	35.9	52.3
RWTH submission (system combination)	36.7	51.8
Jane with all syntactic models (Jan 2011)	37.0	52.0

i.e. with all methods conjoined, the feature vector consists of 27 feature functions. While the scaling factor optimization method Och’s MERT (Section 3.6.2) is known to work only on a small-sized feature vector, the generalization seems to reach a saturation when combining the various methods. For future directions, it should be examined more closely how the methods interact for other optimization techniques like e.g. the MIRA algorithm [Crammer & Singer 03], first applied in SMT in [Watanabe & Suzuki⁺ 07].

Another interesting research direction would be to extend the soft string-to-dependency model so that the language model scoring takes places during decoding, rather than during rescoring. One would need to take care of possible redirection of the dependencies when merging errors occur, since the phrase table is not restricted like in [Shen & Xu⁺ 08]. However, since the language model score is only invoked when the head word is determined, this only poses a minor obstacle.

Sign Language Corpus Analysis

In this chapter, we define an SMT framework for sign languages. Sign languages are natural languages with a grammatical structure and vocabulary that is different from spoken languages. They are not acoustically conveyed with sound patterns, but rather transmitted through visual sign patterns, by means of facial expression, body language and manual communication. Another great difference is the ability to convey information on multiple layers of communication channels simultaneously, compared to the sequential nature of most spoken languages. Sign languages are the primal means of communication for most congenitally deaf¹ persons and many hard-of-hearing persons.

The chapter is organized as follows: in Section 6.1, we will look more closely at notation systems and typical grammar structures in European and American Sign Languages. Section 6.2 focusses on sign language corpora. We finish this chapter by briefly discussing the usefulness of sign language machine translation (SLMT) in Section 6.3.

6.1 Sign Language Grammar

Despite common misconception, each sign language is different from the other. For example, British Sign Language differs considerably from American Sign Language, although the countries share approximately the same spoken language. Even within one country, a huge variety of dialects within the local sign language(s) is common.

In this section, however, we will describe some characteristics which seem to be prevalent in most sign languages. Note that while our main focus is

¹A large proportion of the deaf community actually considers itself to be part of a cultural minority rather than being impaired, differentiating the hearing condition “deaf” from the social group “Deaf” by a capital “D” (cf. [Rexroat 97]). We will adopt this notation in the following.



(a) ASL sign for “LADY” in Stokoe Notation. (b) DGS sign for “ALSO” in Hamburger Notation System.

Figure 6.1: Two example transcriptions of signs in (a) Stokoe notation and in (b) the Hamburger Notation System.

on German Sign Language (German: “Deutsche Gebärdensprache”, DGS) and Sign Language of the Netherlands (Dutch: “Nederlands Gebarentaal”, NGT), some of the references in this section report on other sign languages. We have included them whenever we felt from our own experience that their findings can be carried over to DGS and NGT, but the effects might not apply in their full range. Pictures in this chapter are taken from [Braem 95].

6.1.1 Notation Systems

Sign languages lack an official notation system. Rather, there exist a large quantity of different systems which differ greatly in accuracy and expressiveness, based on their description purpose. One of the earliest systems is the model defined in [Stokoe 60]. It is designed to describe the sign based on the hand configuration, the place of articulation and its movements with an inventory of only 55 phonemes.² For example, Figure 6.1(a) shows a sign transcribed in this notation: the thumb of a spread hand (5) brushes the chin (·) as it moves downward (↓) to touch (x) the breastbone (□). Nowadays, the more versatile Hamburger Notation System (HamNoSys) [Prillwitz 89] is used for the description of the components. For example, in Figure 6.1(b), the index finger (∩) is pointing half from the body half upwards (↑) with the palm in normal position (0) and does a repeated (↻) movement half downwards half further from the body away (⊕).

It should be noted that the above examples in Figure 6.1 only describe the manual components of a sign. Furthermore, they are not particularly useful for MT, since for example the same sign from two different signers might be executed at slightly different body locations and thus be transcribed differently. A better notation form that relies on the semantic representation of a sign is called a *gloss*. As a convention, the meaning of the sign is written as the stem form of the corresponding word in a spoken language, usually in upper case. For a summary of the annotation symbols used, see Appendix C.

²Note that Stokoe himself described the sub-units as *cheremes* in his original article, but this term is hardly used nowadays.

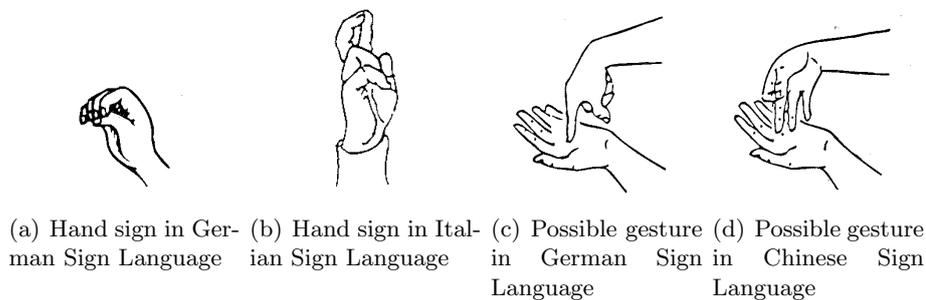


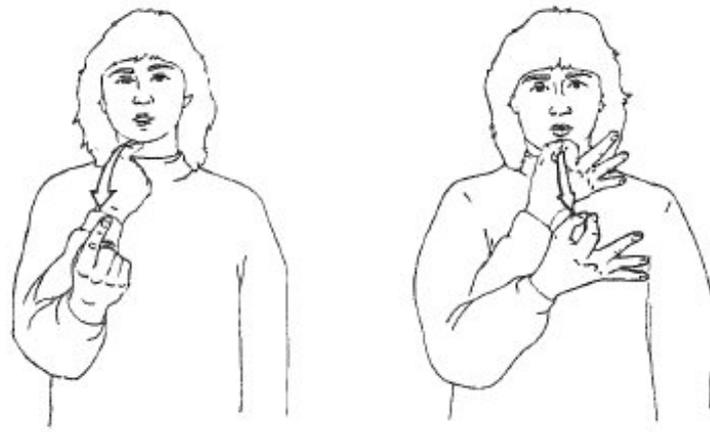
Figure 6.2: Samples for admissible hand configurations and their combination in different sign languages. Figure (a) is a common hand shape that occurs frequently in DGS, whereas Figure (b) is not part of any meaningful phrase but occurs in Italian Sign Language. Likewise, the gesture in Figure (c) is an admissible combination of two hands that occurs in DGS, whereas the gesture in Figure (d) is not, although it appears in the Chinese sign language.

6.1.2 Components

Sign languages convey language information by means of facial expression, body language and manual communication. While the hands offer a considerable amount of versatility and carry a huge portion of the meaning, the non-manual components include crucial information as well. For example, in [Baker & Padden 78] it was shown that it is possible to gain some information of the conversation topic even if the hand of the signing person cannot be seen, which strongly suggests that the non-manual devices go well beyond meta-language information.

Let us review the manual components first. There is a huge variety of different patterns that hands and arms can theoretically form, but similar to lexical units and their allowed combinations in spoken languages, not all of them will appear simultaneously within one language (cf. Figure 6.2). Minimal pairs, i.e. two semantically distinct words that differ in one particular aspect, can be found for the hand configuration, the hand situation, the place of articulation and the movements [Klima & Bellugi 79, Battison 78]. An example of a minimal pair in DGS for the hand configuration is given in Figure 6.3.

There are also many examples for non-manual devices that play a crucial role in a signed sentence. Facial expression and head position can be used to indicate questions, negations and sub-clauses (e.g. [Sandler 99]) or otherwise change the meaning of the sign. Other examples for important non-manuals include the upper part of the body, which can be turned to indicate a role change of the speaker in direct speech, and the lips can be used to dis-



(a) Outstretched index finger (b) Index finger touching the thumb

Figure 6.3: Minimal pair example in DGS. The hand situation, the place of articulation and the movement remains the same, but the hand configuration differs. Figure (a) means “SAY”, Figure (b) means “ASK”.

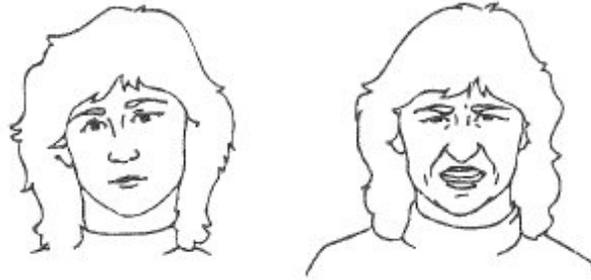
criminate between signs which have the same manual components, specify subordinated signs or carry other additional information. Conditional sub-clauses are indicated in DGS through raised eyebrows and a slight tilt of the head.

6.1.3 Selected Phenomena in Sign Languages

We will proceed to point out some important aspects of sign languages that will have a high impact on the translation. Perhaps the most striking one is that sign languages have a spatial modality. For example, many flexed verbs share the same root, i.e. being mostly identical in their components, but differ in such elements like movement speed, direction or amount of signing space used. For undefined pronouns not present in a conversation, the direction in which a verb sign is executed can specify subject, object and their number of occurrences (cf. Figure 6.5).

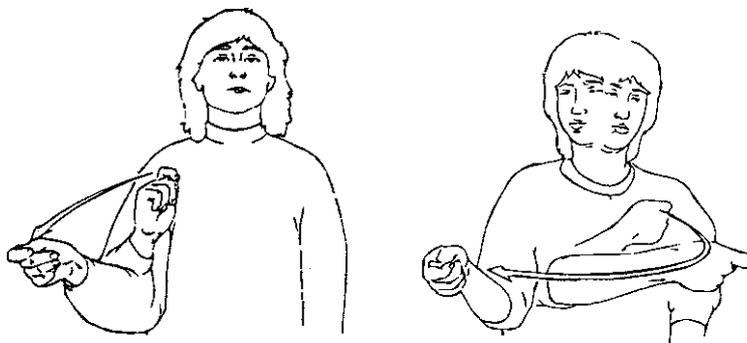
It is also possible to store specific persons or objects in the signing space (cf. [Wrobel 01]), a technique called a *discourse entity*. Later, the entity can be referenced by a pointing sign or a verb, similar to a pronoun (e.g. “*She* is giving the book to *him*.”). While pronouns also occur frequently in spoken languages, sign languages are actually less ambivalent since the object represented by a certain signing space is better defined.

Another important aspect of sign languages is their ability to convey meaning on parallel information channels, as already mentioned above. Non-manual devices can operate more or less independent while signing with the



(a) Neutral facial expression (b) Intensive facial expression

Figure 6.4: Examples for different facial expressions that change the meaning of a sign. For example, a certain gesture meaning “QUESTION” in DGS would have the facial expression as in Figure (a), but would mean “INTERROGATION” if it had the facial expression as in Figure (b).



(a) Verb flexion meaning “I give you” (b) Verb flexion meaning “I give you all”

Figure 6.5: Illustration of two flexed forms of the verb “GIVE”. The sign in Figure (a) is a simple forward movement and means that the signer is giving something to a single person, whereas the sign in Figure (b), sharing the same hand form but executed with a huge arc, means that the signer is giving something to a group of people.

hands, and e.g. a head tilt could indicate a sub-clause, or a head-shake could negate the meaning of a sentence.

6.2 Sign Language Corpora

While there is a considerable amount of material available for sign languages, finding a suitable corpus for sign language translation is still complicated. Often, the collection consists of videos with signed content and its (rough) translation as subtitles, or the collection consists of the videos and the gloss transcription. A corpus with glosses and their translation, however, is rare because of its rather large transcription time overhead. Also, some of these data collections focus on linguistic issues with each phenomenon having only a few sentences, or they have a domain that is too broad to be suitable for machine translation.

In this section, we are presenting the corpora that we will use in the following chapter: the RWTH-PHOENIX and the Corpus-NGT corpus. We will also review other existing corpora and elaborate on the reasons why we did not choose them.

Most of this work has been carried out as part of the European-funded SIGNSPEAK project [Dreuw & Forster⁺ 10b].

6.2.1 Corpus RWTH-PHOENIX

The RWTH-PHOENIX corpus consists of a collection of richly annotated video data from the domain of German weather forecasting. It includes a bilingual text-based sentence corpus and a collection of monolingual data of German sentences. This domain was chosen since it is easily extendable, has a limited vocabulary and features real-life data rather than material made under lab conditions. It was first described in [Bungeroth & Stein⁺ 06], and extended in [Stein & Forster⁺ 10]. The older recordings differ from newly recorded videos, since the television programme has changed in two important aspects. First, the format of the video is different: before, the news announcer was slightly distorted in perspective, and the signing interpreter was shown without a background of its own. Now, the broadcast channel shows the original video in a smaller frame and places the signing interpreter in front of a grey background on the far right (cf. Figure 6.6). For MT, this does not pose a problem since the algorithms only work on the transcriptions and not on the video signal. Recognition of signing in a video is beyond the scope of this work, instead, we refer the reader to [Stein & Dreuw⁺ 07].

As for the second major change in the data, the transcription of the audio material is no longer provided by the broadcast station. We therefore employ an automatic speech recognition system for the German audio data which transcribes the spoken words, and manually align the words to the

annotated gloss sentences. For the weather forecast, the audio recognition word error rate is below 5%, making the transcription quite convenient.

Quality and Usability

Although the interpretation of the news announcements into German Sign Language is performed by bilingual experts, the translation quality suffers from the recording situation: the interpreters have to listen to the announcements under real-time conditions and thus have to sign simultaneously without any preparation, since they do not receive a transcript in advance. Due to the complex nature of official news announcements and the relatively high speed of the announcer, the signed sentences are generally in German Sign Language, but in some sentences there is a slight bias towards the grammatical structure of spoken German (cf. Figure 6.7). However, a manual inspection showed that such bias only occurs rarely. A glitch which is more common in the corpus is the omission of details in the signed sentences. For example, if the announcer talks about the region of Bavaria, the adjacent Austrian Alps and the river Donau, the interpreter might more generally refer to the south of Germany without specifically naming the exact locations. Another typical omission occurs when the announcement refers to specific wind velocities such as “light”, “gentle”, and “fresh”, and the interpreter only differentiates between a low and a high velocity. Since the translation errors occur only occasionally and are not consistent, this makes the translation task actually more difficult, because the sentence pairs contain some information mismatch, which leads to errors in the word alignment.

A different kind of mismatch is introduced by the gloss-notation system. Sometimes, the interpreter expresses certain aspects of the spoken sentence by modifying a sign or by non-manual means such as facial expression or body posture. These aspects are not always captured in the glosses. Note that the choice of the gloss labels in this corpus was not influenced by the text of the spoken language. Our Deaf colleague who conducted the annotation did not refer to the text spoken by the announcer but only used the videos showing the signing interpreters.

German Sign Language uses the repetition of signs to express plurality of nouns, distributive aspects, etc. Such phenomena can be annotated in different ways, depending on the focus and use of the corpus. In the RWTH-PHOENIX corpus, if a sign is repeated a limited number of times, e.g. 2-3 clouds, each sign is annotated individually. However, we do use a special suffix “++” to indicate the quick repetition of a sign which implies a new meaning (e.g. the DGS gloss “QUESTION++” which means “interrogating” rather than “asking”). No special annotation was used in the first case, since the corpus is also used to train the sign language recognition system, for which individual annotation of each repetition is more appropriate, since a special indicator would imply that different models have to be trained for



Figure 6.6: Old and new television format used in the Phoenix television channel

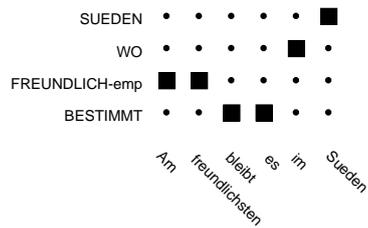
singular and plural nouns, which is not feasible given the current corpus size. Since the number of repetitions often varies, the computer sometimes translates the same word several times instead of translating the noun as a plural or using another appropriate expression in the spoken language.

The mismatch between the information contained in the glosses and in the text spoken by the announcer leads to several problems when training a statistical machine translation system. Since the words omitted by the interpreter have no correspondence in the glosses, the automatic word alignment system either aligns these words to other unrelated glosses or does not align them at all. While the former leads to wrong translations, the latter leads to the omission of these unaligned words, which means the translation system cannot even reproduce the correct translation of sentences it has seen during training.

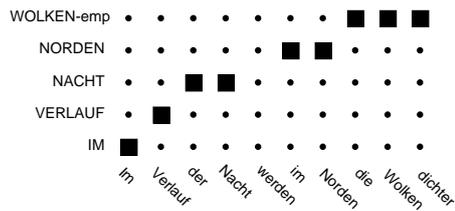
Annotation

For the annotation, we employ the ELAN tool [Brugman & Russel 04], which is widely used for various sign language corpora (e.g. the ECHO corpus [Crasborn & van der Kooij⁺ 04]) and extended to the special needs of this language modality [Crasborn & Sloetjes 08].

Both left-hand and right-hand movements are kept track of independently, but annotated into the same stream of words. Our annotator is congenitally deaf and has worked in research fields regarding sign language for over a decade, but had no previous annotation experiences. According to his feedback, it took him about two weeks to become accustomed to the annotation tool. For the first two months working on the recordings, various questions arose concerning the annotation procedure, namely for such effects as dialects, synonyms, classifiers, left-hand/right-hand issues which were discussed in his mother tongue with interpreters. At first, it took him four hours to annotate one weather forecast of roughly one minute. After two months, he was able to finish three videos in the same amount of time. For the whole news announcement, which basically has a unlimited domain



(a) Correct translation.



(b) Translation that seems to be influenced by the German grammar. A more correct translation would be e.g. “NACHT VERLAUF WOLKEN-emp WO NORDEN”.

Figure 6.7: Example translations taken from the RWTH-PHOENIX corpus. Sentence (a) features a rhetoric question (literal translation: “The most friendly weather remains where? In the south.”), a common grammatic device in sign languages. Sentence (b), however, is quite close to the German grammar and can be considered to be a signing glitch by the interpreter.

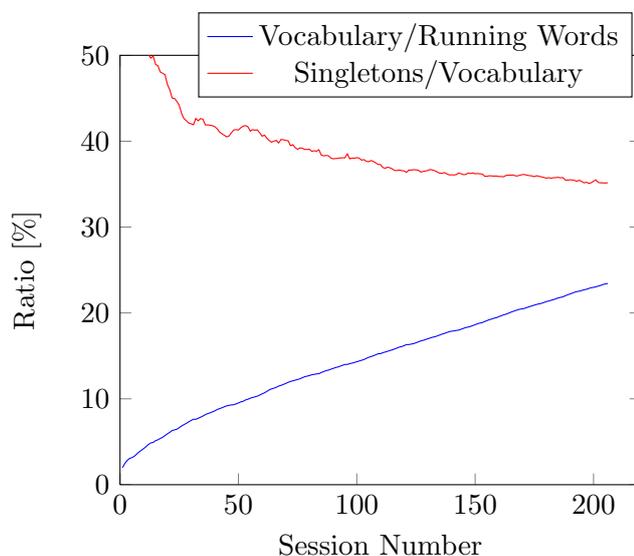


Figure 6.8: RWTH-PHOENIX: ratio of the vocabulary size compared to the running words, and ratio of the singletons compared to the vocabulary.

and runs for 15 minutes, it takes him about 24 working hours to transcribe it.

Corpus Progression

For a complete corpus overview, see Table 6.1. Comparing the corpus statistics with other small-sized data selections, the selected domain seems to offer suitable progress based on the rather tedious and slow annotation process. For example, the Chinese–English task of the International Workshop on Spoken Language Technology (IWSLT)³ is a selection of parallel sentences in the domain of travel and booking information, has 22 K training sentences, with a type-token ratio (i.e. the average number of running words per vocabulary entry) of 18.8 for Chinese and 27.5 for English. Compared to our corpus, we currently have a total of 2.7 K training sentences and already approach a type-token ratio of around 20 (cf. Figure 6.8) after 220 sessions. The singleton ratio is about 40% for both languages in IWSLT, while ours drops quickly below 35% and seems stationary. As can be seen in Figure 6.9, peaks in the vocabulary growth (likely to contribute to the singleton and out-of-vocabulary ratios) can mostly be attributed to time-specific terms like special seasons or certain places where weather phenomena occur in a certain week. Since these words tend to occur often in consecutive sessions, the singleton ratio typically drops fast.

³<http://mastarpj.nict.go.jp/IWSLT2009/>

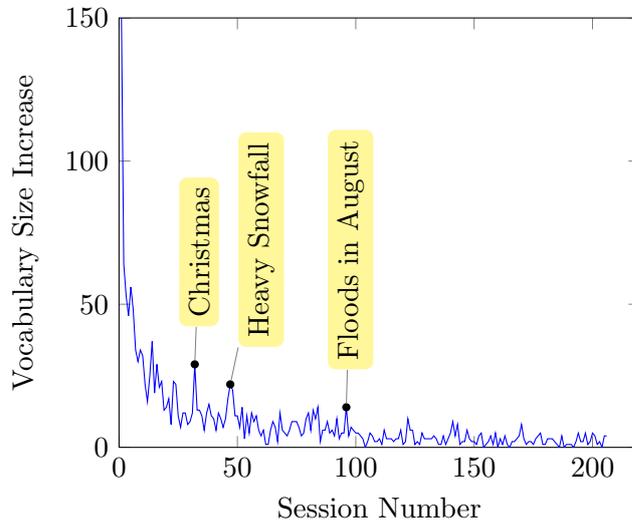


Figure 6.9: RWTH-PHOENIX: vocabulary size increase per session and notable topics of the specific broadcasts.

Table 6.1: Corpus statistics for the weather forecast corpus RWTH-PHOENIX (Section 6.2.1).

		Glosses	German
Train:	Sentences	2565	
	Running Words	31 208	41 306
	Vocabulary	1 027	1 763
	Singletons	371	641
Test:	Sentences	512	
	Running Words	6 115	8 230
	Vocabulary	570	915
	OOVs	86	133
	Trigram ppl.	51.7	22.7

Table 6.2: Corpus statistics for Corpus-NGT. The test set consists of multiple references (2–4, 2.5 on average). Here, only the statistics for the first sentence in each reference are presented.

		Right Hand	Left Hand	Dutch
Train:	Sentences		1699	
	Running Words	8 129	4 123	15 130
	Vocabulary	1 066	773	1 695
	Singletons	481	376	840
Test:	Sentences		175	
	Running Words	875	496	1 815
	Distinct Words	272	181	426
	OOVs	46	39	39
	Trigram ppl.	107.0	54.6	67.5

6.2.2 Corpus NGT

[Crasborn & Zwitterlood 08] presents a data collection in the Sign Language of the Netherlands. It consists of recordings in the domain of fable, cartoon/home video paraphrases, discussions on various topics and free conversation. A translation into spoken Dutch, however, was missing at first, and we had to decide on which sub-part to focus first. After a careful scan of the data, we excluded the topics “funniest home videos”, fables and “Tweety & Silvester” because of their huge amount of iconographic signing, a daunting task for both recognition and translation. In addition, we discarded free conversations as well as talks about self experiences because they only had an average type-token ratio of 3.2 and 4.8, respectively. The domain of discussions on selected topics that are related to deafness and Deaf culture, in contrast, had an average type-token ratio of 8.5 and 6.0, since the vocabulary was somewhat restricted due to the specific questions that the signers were arguing about. In this setting, two signers are sitting face to face and discuss a topic that was shown to them in form of a written question. The translations of the sentences were provided by the Radboud University, Nijmegen, as part of the shared EU-project SIGNSPEAK.⁴

Quality and Usability

The Corpus-NGT can be considered to be far more challenging than the RWTH-PHOENIX corpus. With its current data size (cf. Table 6.2), we do not expect to reach satisfying translation results. This is mainly due to

⁴<http://www.signspeak.eu/>



Figure 6.10: Screenshots from the Corpus-NGT corpus (Section 6.2.2)

right hand	MOEILIJK DOEN OVER	COMMUNICEREN PO	MET	IX HOREND	MENSEN PO
left hand	MOEILIJK DOEN	COMMUNICEREN	MET	MENSEN	
Dutch	Erg veel moeite doet om te communiceren met horende mensen.				

(a) “It is quite hard to communicate with hearing persons”

right hand	ALS IX-1 LANG NIET	BETEKENEN EMAIL	BETEKENEN GEBAREN IX-1 PO
left hand	NIET HEEN CLUBHUIS TOE	BETEKENEN	GEBAREN IX-1 PO
Dutch	als je lang niet naar het clubhuis gaat , weet je het gebaar voor het woord e-mail bijvoorbeeld niet .		

(b) “If you haven’t been to the club house for some time, you will miss the sign for the word ‘email’ ”

Figure 6.11: Example sentences from the Corpus-NGT corpus. Each hand is annotated with a separated tier.

the much broader domain when compared to weather forecasts, and due to the conversational, even casual nature of the signed sentences. Hesitations and partial sentences are frequent, and some information is only conveyed by non-signed (and thus non-glossed) communication channels like facial expressions (e.g. “I totally agree”), while still translated into Dutch. However, the corpus is also more interesting from a scientific point of view for the following reasons. First, we can assume the grammar of the sign language to be more accurate, since it is derived from close-to-natural conversations among Deaf. Apart from the written questions that start the topic discussions, the Dutch grammar will not have an immediate impact on the word order or the communication devices used. Second, the annotation procedure reflects the parallel nature of sign languages in a better way. The corpus features two different annotation tiers for each hand (cf. Figure 6.11), and we also have time-aligned head shake annotations. While neither of these circumstances will ease our task, the corpus still feels like the next logical step for broad domain SLMT.

6.2.3 Other Corpora

Apart from these two, there are other corpora available. The European Cultural Heritage Online organization (ECHO) published data collections for Swedish Sign Language, British Sign Language and Sign Language of the Netherlands [Crasborn & van der Kooij⁺ 04]. Their broad domain of children’s fairy tales as well as poetry make them rather unsuitable for statistical methods. Another obstacle is the intensive usage of signed classifiers because of the rather visual topics.

Another data collection for Czech and Signed Czech is presented in [Kanis & Zahradil⁺ 06]. Its domain is taken from transcribed train timetable dialogues and then translated by human experts. However, the actual translations are not in the Czech Sign Language spoken by the Deaf, but in an artificial language system strongly derived from spoken Czech. Explicit word alignments are made by human experts. Due to its nature, the authors are able to achieve very high performance scores, as already pointed out in Section 6.3.1. It does not appear to be the most challenging data collection for SMT.

[Bertoldi & Tiotto⁺ 10] recently announced that they have started to build up a sign language corpus that is also in the domain of weather forecast, thus quite similar in content to the RWTH-PHOENIX, but for Italian and Italian Sign Language. However, their annotation procedure differs considerably. Lacking publicly available signed interpretation of the news broadcast material, they employ a speech recognizer instead and have the spoken language transcription interpreted by a bilingual expert. The same expert then transcribes the glosses of his own signing captured in a video recording. [Massó & Badia 10] work on weather forecast as well, for Catalan and Catalan Sign Language, but report problems when tuning the SMT system due to their overall corpus size of 154 sentences. We will review their efforts in Section 7.4.1.

The Air Travel Information System (ATIS) is a corpus for English, German, Irish Sign Language, German Sign Language and South African Sign Language described in [Bungeroth & Stein⁺ 08]. With roughly 600 parallel sentences in total, it is small in size. However, being a multilingual data selection, it enables direct translation between sign languages. Its performance for SMT has already been extensively studied in [Morrissey 08].

6.3 Usefulness of Sign Language Machine Translation

The phenomenon of lower literacy skills within the deaf population is often mentioned in SLMT papers. Nevertheless, due to new educational methods and a changing awareness (e.g. the acknowledgment of German Sign Lan-

guage by the German government in 2002), we find it worthwhile to look at more recent papers on the current situation. We also perform a small sanity check to see whether statistical machine translation is an appropriate technique for the translation of the corpora at hand.

In general, deaf people perform on par with hearing persons when tested on general intellectual skills that do not require additional (school-)knowledge, as could be expected (see [Kramer 07]), as well as on (sign) language development when growing up in deaf families. However, even the most recent surveys, e.g. [Becker 10] for Germany and [Wauters & van Bon⁺ 06] for the Netherlands, point out huge discrepancies in the writing and reading skills of deaf children when compared to their hearing peers. The authors in [Hermans & Knoors⁺ 08a] recall that, among other possible reasons, the hearing loss of a deaf child impedes its phonological awareness, an important prerequisite for hearing children when learning the alphabetic principle. Further, since the vocabulary in a spoken language is often limited as a result of the restricted access to sound, learning the natural relations between the spoken and a written form of a language is an additional obstacle. The same authors suggest in [Hermans & Knoors⁺ 08b] that no substantial improvement in almost two decades of bilingual education have been seen so far, despite the finding that proficiency in sign language correlates with reading proficiency. They argue that existing learning models might still not be accurate enough to really support young deaf pupils sufficiently, suggesting that a more direct, explicit link between existing sign language knowledge and new written texts might be necessary.

Even from this short glance at the recent literature it seems safe to conclude that reading and writing in a spoken language is still a huge challenge for many deaf people and will remain so for many years to come. Since sign languages are the most accessible language to many deaf people, translation systems which translate from a video of the signed utterance into the spoken language or featuring an avatar which signs the translation of the spoken language utterance may prove beneficial once a certain translation quality has been established.

6.3.1 Sanity Check: Lower Case Translation

In a recent paper [Kanis & Müller 09], the authors worked on the translation of an intermediate, signed form of the Czech language and obtained translation results of up to 81 BLEU, which is probably due to the similarity between this hybrid language and written Czech. We examine whether such a similarity is also true in the case of German Sign Language, anticipating some baseline results in the next chapter. In parallel to a standard SMT approach, we perform a sanity check to see whether the MT process is actually necessary and helpful.

Since sign languages are typically transcribed as glosses that are repre-

Table 6.3: Results for the sanity check (Section 6.3.1). Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$.

	BLEU	TER	PER
simple lower casing	2.1	85.7	81.5
4-letter stems	2.6	81.1	74.8
MT system	22.0	74.0	65.1

sented as upper-case words of the corresponding spoken language, a casual viewer might question whether the glosses could simply be written in lower-case letters to generate an acceptable output. To show that the languages differ considerably, we compute the translation metrics on the lower-cased glosses of the RWTH-PHOENIX corpus. Since the glosses are not conjugated nor inflected in the usual way, we also tried to make a fairer comparison by eliminating the inflection using fake word “stems”: each word in the hypothesis and the reference was truncated to its first four letters. As a comparison, we set up a hierarchical phrase-based translation system as a baseline. The system was trained on 2000 sentences and optimized on a development set that was separated from the training data.

The results can be found in Table 6.3. As expected, the SMT system clearly performs better than simply lowercasing the glosses. For a better interpretation, we also included the position-independent error rate (PER) in the table, which is defined as the percentage of words that were translated incorrectly, regardless of their position in the sentence.

6.4 Conclusion

In this chapter, we analyzed some existing data collections and emphasized their individual strength and weaknesses based on established measures like overall size, type-token ratio, and trigram perplexity. We also highlighted specific challenges due to the annotation procedure of a visually conveyed language. For the translation experiments in the next chapter, we picked the RWTH-PHOENIX corpus, which is one of the largest data collections available, and further introduced the Corpus-NGT, featuring a broad domain and parallel input tiers. Arguing that SLMT is feasible and useful, we will perform several experiments in the next chapter that aim at increasing the translation quality.

Sign Language Translation

Sign languages represent an interesting niche for SMT that is typically hampered by the scarceness of suitable data, and most papers in this area apply only a few well-known techniques and do not adapt them to small-sized corpora. In this chapter, we will propose new methods for common approaches like scaling factor optimization and alignment merging strategies. We also conduct experiments with different decoders and employ state-of-the-art techniques like soft syntactic labels as well as trigger-based and discriminative word lexica and system combination.

This chapter is organized as follows. First, we give a short overview of related work in Section 7.1. Then, in Section 7.2, we briefly list the specifics of the translation systems, the system combination and the alignment merging strategies that are employed throughout this chapter.

In Section 7.3, we will focus on the preparation of the data and how this impacts translation quality. We will compare the phrase-based and the hierarchical translation systems and see what can be learned from their individual strengths and weaknesses. We will also introduce methods for preparing morphologically complex languages like German as the source language, and lastly propose strategies for preparing a sign language corpus which features both hand movements as individual input streams.

In Section 7.4, we will perform various experiments for the translation from German Sign Language into spoken German, and review the parameter optimization. We will discuss techniques which are an alternative to withholding a development set when optimizing the scaling factors, analyze the common error measures on this language pair and perform experiments that are syntactically motivated. We also describe models to create suitable translations with a system combination setup.

7.1 Related work

In this section, we give a brief overview of related work in the field of sign language machine translation. Moreover, we discuss some of the related work in the following sections where we directly compare our results and findings to those mentioned in the literature, especially in cases where we disagree with other authors.

Early analysis of machine translation of sign languages were mainly rule-based [Veale & Conway⁺ 98, Sáfár & Marshall 01]. First ideas on statistical machine translation of sign languages were presented in [Bauer & Kraiss 01, Bungeroth & Ney 04], but these papers do not offer many experimental results. Recent works in this area include:

[Morrissey & Way 06] report problems when using the standard error measures in sign language translation. They point out that for small test sets and for unstable data, BLEU is a bad choice as an optimization metric, since e.g. sometimes no correct four-gram can be found. The same authors report in [Morrissey & Way⁺ 07] that in more recent experiments the BLEU scores on the ATIS corpus [Bungeroth & Stein⁺ 08] are on reasonable levels again, but leave the question open whether this is due to better data or better machine translation systems. The main author gives an in-depth investigation of corpus-based methods for data-driven sign language translation in [Morrissey 08].

A system for the language pair Chinese and Taiwanese sign language is presented in [Chiu & Wu⁺ 07]. The authors show that their optimization method surpasses IBM model 2, but leave the question open how their system reacts on other translation models.

In this article, we will review two more recent papers more closely: [Kanis & Müller 09] work on the translation of an intermediate, signed form of the Czech language and obtain translation very high results, which probably is due to the similarity between this hybrid language and written Czech.

[Massó & Badia 10] uses factored models on a standard phrase-based system for Spanish to Spanish Sign Language. They report that their data behaves unexpectedly during optimization in that they achieve the best results on their test set if they use the complete training set for the estimation of the scaling factors.

Two larger projects focusing on corpus generation and translation techniques have emerged recently: the SIGNSPEAK project as described in [Dreuw & Forster⁺ 10a], in which the authors take part, aims at combining scientific theory on translation and vision-based technology on image recognition within a common framework, including linguistic research. The goal is an overall sign language recognition and translation system. The other project, called DICTASIGN [Efthimiou & Fotinea⁺ 09], aims at developing Web 2.0 interactions in sign language. Signing is recorded via webcam, recognized and then represented with an animated avatar. Anonymization as

well as sign language to sign language translation is within possible application range.

7.2 System Description and Preliminaries

In this chapter, we use both a phrase-based and a hierarchical phrase-based decoder. In the following, we will describe the systems we used.

7.2.1 PBT: Phrase-based Translation

We used an in-house phrase-based translation system called PBT, as described in [Zens & Ney 08]. Different models are integrated into a log-linear framework (see Eqn. 3.16).

The models h_m used in the phrase-based translation system are: phrase and word translation probabilities in both directions, a standard n -gram language model, word penalties, phrase penalties, distortion penalties and a discriminative reordering model.

7.2.2 Jane: Hierarchical Phrase-based Translation

We also used Jane, as described in Chapter 4. The following models were used for the baseline: translation probabilities, IBM-like word lexica, word- and phrase penalty as well as binary markers for various hierarchical phrases.

7.2.3 System Combination

For system combination, we use an in-house system which has been extensively described in [Matusov & Leusch⁺ 08]. Here, we compute a weighted majority voting on a confusion network, similarly to the well-established ROVER approach [Fiscus 97] for combining speech recognition hypotheses. To create the confusion network, pairwise word alignments of the original MT hypotheses are learned using an enhanced statistical alignment algorithm that explicitly models word reordering. Instead of only considering a single sentence, the context of a whole corpus is taken into account in order to achieve high alignment quality. The confusion network is rescored with a special language model, and the consensus translation is extracted as the best path in it.

7.2.4 Alignment Merging Strategies

In this chapter, we will analyze the impact of the alignment on the translation quality. If the alignments are computed with the IBM Models (see Section 3.2.2), the result differs based on the translation direction, which is

why the standard alignment is typically merged heuristically with the inverse alignment. Let $\mathcal{A}_{f \rightarrow e}$ be the alignment for the standard source-to-target direction, and let $\mathcal{A}_{e \rightarrow f}$ be the alignment for the inverse direction. The easiest merging strategies to compute are the intersection $\mathcal{A}_{\text{intersection}} = \mathcal{A}_{f \rightarrow e} \cap \mathcal{A}_{e \rightarrow f}$ and the union $\mathcal{A}_{\text{union}} = \mathcal{A}_{f \rightarrow e} \cup \mathcal{A}_{e \rightarrow f}$. Many algorithms try to find an intermediate alignment between these extremes by starting with the intersection alignment and then merging it with a selected set of appropriate alignment points taken from the union alignment. The most common is the *grow-diag* algorithm as presented in [Koehn & Och⁺ 03]. It extends iteratively every alignment point whenever it has a direct neighbor, i.e. when a vertically, horizontally or diagonal adjacent alignment point is in the union alignment but not yet part of our merged alignment (see Algorithm 4). An alternative to this approach is presented in [Och & Ney 03b], where the possible neighbours are restricted to vertically and horizontally adjacent positions. In addition, alignment blocks are avoided by only allowing extension in one direction at a time. We will denote this method as *grow-mono* (see Algorithm 5).

One downside to the restriction to adjacent neighbours is that we cannot reach far-off, isolated words that remain unaligned in the intersection alignment. Many merging strategies therefore employ a second run called *final*, where we iteratively insert alignment points into so-far non-aligned rows and columns. With *final-and*, we denote a similar strategy that makes sure that both row and column must be free. Algorithm 6 illustrates the differences between these approaches. [Och & Ney 03b] also employ a strategy where they start a *final-and* first and then expand this alignment with the *grow-mono* algorithm, a strategy which we will denote as *final-and-grow-mono*.

In our experiments, we will analyze the performance for the *union* alignment, the *intersection* alignment, the *grow-diag-final-and* alignment, and the *final-and-grow-mono* alignment.

Algorithm 4: Grow-Diag

```

input : alignments f2e and e2f
output: merged alignment new-alignment
neighbors :=
[(-1, 0), (0, -1), (1, 0), (0, 1), (-1, -1), (-1, 1), (1, -1), (1, 1)];
new-alignment = f2e  $\cap$  e2f;
while new alignment points are inserted do
  for each ap in new-alignment do
    for each np in neighbors(ap) do
      if (np  $\in$  f2e  $\cup$  e2f) and (np  $\notin$  new-alignment) then
        new-alignment.insert (np);

```

Algorithm 5: Grow-Mono

input : alignments f2e and e2f
output: merged alignment new-alignment
new-alignment := (f2e \cap e2f);
vertical-neighbors := [(-1, 0), (1, 0)];
horizontal-neighbors := [(0, -1), (0, 1)];
while *new alignment points are inserted* **do**
 difference-alignment := (f2e \cup e2f) / new-alignment;
 for each ap **in** difference-alignment **do**
 vertical := \exists vertical-neighbors (ap) \in new-alignment;
 horizontal := \exists horizontal-neighbors (ap) \in new-alignment;
 if ((vertical **and** not horizontal) **or**
 (horizontal **and** not vertical)) **then**
 └ new-alignment.insert (ap);

Algorithm 6: Final-(And)

input : alignments new-alignment and unify-alignment
output: finalized alignment new-alignment
for each ap **in** unify-alignment **do**
 if ap \notin new-alignment **then**
 if new-alignment.row (ap).free (**and/or**)
 new-alignment.column (ap).free **then**
 └ new-alignment.insert (ap);

7.2.5 Evaluation

As before, all results are reported in terms of BLEU [Papineni & Roukos⁺ 02] and TER [Snover & Dorr⁺ 06]. Their improvements over the baseline (usually reported in the first line of the result table) are checked for statistical significance, using pairwise bootstrap [Koehn 04] with 500 random samples.

7.3 Preparation of Sign Language Corpora

Following the analysis of the corpora in the last chapter, we will now focus on the preparation of the data and how it impacts translation quality.

In Section 7.3.1, we will compare the phrase-based and the hierarchical translation system and see what we can learn from their individual strengths and weaknesses. We noted in [Stein & Schmidt⁺ 10] that the phrase-based system outperformed the hierarchical system on most tasks and will pinpoint some possible reasons.

In Section 7.3.2, we will review a linguistic tool-based method for splitting compound words when translating from a morphologically rich spoken language.

In Section 7.3.3, we will propose strategies for preparing a sign language corpus which features both hand movements as individual input streams.

7.3.1 Translation Paradigm Comparison

Compared to [Stein & Schmidt⁺ 10], where we ran experiments on the same RWTH-PHOENIX data, both the phrase-based and the hierarchical systems improved considerably. For the phrase-based system, we introduced more sophisticated categories for ordinal numbers and dates, which occur frequently in the weather domain.

In the same set of experiments, the hierarchical system performed noticeably worse than the phrase-based system. After carefully checking the hypotheses, we noticed two sources of errors. First, the categories in the phrase-based decoder were filtered by default such that no rule contains a different number of categories in the source and the target phrase, which can happen due to the extraction procedure. We adopted this system in the hierarchical extraction as well. The second source of error was due to sentence-end markers. In the gloss annotation, sentence end markers in the sign language are missing, but the German sentence often ends with a full stop. While the left-to-right search strategy in the phrase-based system seems to have fewer problems in producing a full stop, the search algorithm in the hierarchical system often failed to do so. We thus added an artificial sentence-end marker to each gloss sentence, retrained the alignments and optimized the hierarchical system again. The results are shown in Table 7.1.

Table 7.1: RWTH-PHOENIX results for the hierarchical phrase-based decoder and the phrase-based decoder on a union alignment (Section 7.3.1). Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$.

	BLEU	TER
Old Phrase-based System	21.6	68.7
+ Categorization	24.3	65.7
+ Sentence End Markers	24.1	64.8
Old Hierarchical System	19.9	69.7
+ Categorization		
& Phrase Table Filtering	23.8	69.3
+ Sentence End Markers	24.2	67.4

In terms of BLEU, the hierarchical system now performs on par with the phrase-based system for the union alignment.

We proceeded to compare the different alignment strategies (see Section 7.2.4). To obtain alignment quality measures, we hand-aligned 400 sentences to compute precision, recall, F-Measure and alignment error rate (AER) [Och & Ney 00]. See Table 7.2 for a quality estimation of the various merging strategies on these sentences. Especially the intersection alignment performs very poorly for all measures except the precision score (as could be expected), but for the other three alignments, the F-measure and AER are comparable whereas precision and recall are quite different. However, the phrase-based system was hardly affected by a low recall value (see Table 7.3) whereas the hierarchical system showed a significant drop in performance both in terms of BLEU and TER. As it turned out, this was partly caused by the applied extraction heuristics. By default, both systems extract standard phrase blocks (Fig. 7.1(a)), extend them at the border if there are unaligned words (Fig. 7.1(b)) and further extract all alignment dots as word pairs (Fig. 7.1(c)). However, the phrase-based system further extracts word pairs whenever neither source nor target are aligned (Fig. 7.1(d)). When activating this heuristic for the hierarchical system as well, the intersection alignment performance improved from 22.0 BLEU and 77.1 TER to 22.8 BLEU and 72.9 TER.

7.3.2 Translation from a Morphologically Complex Spoken Language

This section deals with the preprocessing for the translation from a written transcription of spoken German into German Sign Language, which would

Table 7.2: Precision, recall, F-Measure and alignment error rate (AER) of the different alignment merging strategies (Section 7.3.1)

	Precision	Recall	F	AER
Union	40.9	62.0	49.3	50.6
Intersect	80.1	28.8	42.0	57.6
Grow-diag-final-and	44.2	55.2	49.1	50.8
Final-and-grow-mono	47.6	45.4	46.5	53.5

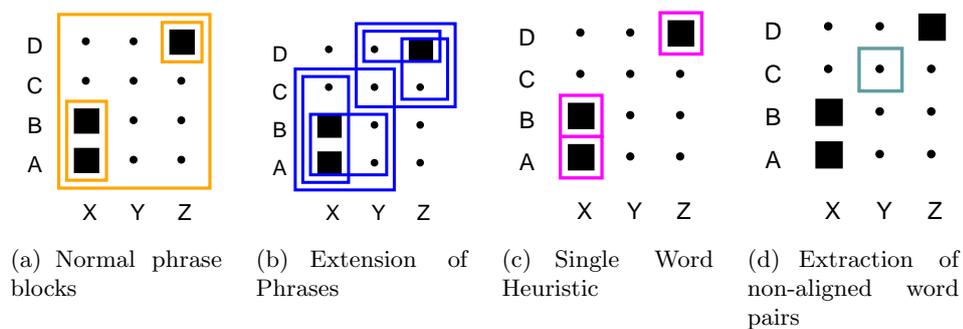


Figure 7.1: Extraction heuristics applied for initial phrases

Table 7.3: RWTH-PHOENIX results for the hierarchical phrase-based decoder on the alignment merging strategies (Section 7.3.1)

	PBT		JANE	
	BLEU	TER	BLEU	TER
Union	24.4	65.7	24.3	67.3
Intersect	24.6	65.1	22.8	72.8
Grow-diag-final-and	24.3	64.9	23.9	68.8
Final-and-grow-mono	24.6	64.9	23.7	69.4

be the first step in an avatar-supported speech-to-sign architecture. For this, we use the same corpus as in the previous section, but reverse the translation order.

The best hierarchical baseline system is derived from a grow-diag-final-and alignment and yields scores of 16.3 BLEU and 76.1 TER. It contains sentence-end markers on the target side, but they are only used during optimization and removed automatically before comparing the hypothesis to the reference. For the phrase-based decoder, the best system is derived from an intersection alignment and yields scores of 15.3 BLEU and 77.0 TER. Adding DWLs and Triplets further improves the performance to 15.5 BLEU and 76.1 TER.

Compared to the results of translations in the other direction, the translation quality is noticeably worse. Possible reasons for this are the rather mixed grammar structure on the sign language part, which can be seen in the trigram perplexity (see Table 6.1, p. 71). Furthermore, some words in the sign language are repeated several times to either emphasize a part of a sentence or to indicate a plural, whereas the exact number of repetitions sometimes seems arbitrary.

Another issue for the translation system is the case of morphologically rich languages such as German as the input language, where especially compound words and inflected nouns pose a problem. We therefore reduced the morphological complexity of the German source language by automatic means. To achieve this, we parsed the data with a morpho-syntactic analysis tool before the actual translation phase. The freely available tool Morphisto¹ [Zielinski & Simon 08] is based on a finite-state transducer with a large database of German words, accurately reporting part-of-speech tags, gender, case and possible split points for large compound words. However, in the case of ambiguous split points it does not provide probability scores for the various possible parsings. We therefore opted to always take the entry consisting of the fewest split points possible. By doing so, we reduce all words to their stem form and split large words automatically.

In Table 7.4, the results for this task are presented. The improvement in BLEU is statistically significant for the phrase-based system, but there is a slight deterioration in the hierarchical system. In both systems, TER improves statistically significant.

7.3.3 Translating from Two Input Streams

In this section, we first will perform experiments on the translation direction from NGT into spoken Dutch. The main problem that we try to address is that of the parallel input channels, because the glosses are independently annotated for each hand.

¹<http://www1.ids-mannheim.de/lexik/textgrid/morphisto/>

Table 7.4: RWTH-PHOENIX results for both decoders for the translation direction spoken German to German Sign Language (Section 7.3.2)

	PBT		JANE	
	BLEU	TER	BLEU	TER
Baseline	15.4	77.0	16.3	76.1
Split Input	15.9	75.7	16.1	74.1

While certainly more accurate, the annotation procedure of glossing each hand independently presents a challenge for the translation system. The example in Figure 6.11(a) (p. 73) shows that for some sentences, the dominant hand covers all words of the sentence and the non-dominant hand remains motionless for signs that only require one active hand. However, this is not always the case. The example in Figure 6.11(b) shows the transcription of a signer who switches the active signing hand within one sentence.

We perform three experiments. First, we only employ the *right hand* information as our source input data and define this as our baseline. The problem of this approach is obvious, since from the personal information of the signers we know that some are left-handed, and an even larger portion switch the dominant hand in between (see Table 7.5). The next approximation is to select for each sentence the glosses of the hand that signs more words, an approach which we call *active hand*. In a third step, we parse the annotation file again and match the timing of the individual glosses, and time-align both gloss tiers, omitting word duplications whenever both hands sign the same (*merged hands*). Note that even if this is the case, the time boundaries for the glosses can differ greatly, e.g. when a signer signs “NEWSPAPER” with both hands, keeps the non-dominant hand in position but signs “COFFEE” with his dominant hand in the meantime, a signed construction which could be translated to “drinking coffee while reading the newspaper”. See Table 7.6 for a quantitative overview of these methods.

The results can be found in Table 7.7. Switching from the right hand to the active hand gives a statistically significant improvement of 1.4 absolute (1.8% rel.) in the TER score, and the merged hand approach further improves the BLEU score by statistically significant 1.9 absolute (22.9% rel.). While in general the translation quality is still low, we expect to gain overall better results with more data, and consider these results as a first step for SLMT in a broader domain.

Table 7.5: Dominant hand of all signers in the Corpus-NGT videos

Left Hand	5
Right Hand	59
Both Hands	14
Unknown	14

Table 7.6: Statistics for the left/right hand merging strategies of the Corpus-NGT corpus (Section 7.3.3)

		Glosses		
		Right Hand	Active Hand	Merged Hands
Train:	Sentences		1699	
	Running Words	8 129	8 625	9 679
	Vocabulary	1 066	1 099	1 175
	Singletons	481	489	481
Test:	Sentences		175	
	Running Words	875	967	1 227
	Distinct Words	272	282	296
	OOVs	46	48	56

Table 7.7: Corpus-NGT Jane results of different strategies for using two gloss streams (Section 7.3.3)

	BLEU	TER
Right Hand	8.1	79.2
Active Hand	8.3	77.8
Merged Hands	10.2	77.4

7.4 Optimizing Sign Language Machine Translation

Following the previous section, which concentrated on the preprocessing of the sign language data, we will focus on the optimization of the translation procedure. The experiments in this section are conducted on the translation direction from German Sign Language into spoken German, which would be the input of a video recognition system in an overall sign-to-speech architecture.

We begin by reviewing the standard optimization procedure of estimating the feature weights in the log-linear feature combination by selection of a proper development set, in Section 7.4.1.

In Section 7.4.2, we will review the choice of the error measure and examine whether the standard approach of optimizing on BLEU is actually the best choice. We compare the system optimized on BLEU with the second-best system with human evaluation.

In Section 7.4.3, we perform syntactically motivated experiments. In [Stein & Schmidt⁺ 10], advanced models employing syntactic parser information did not yield any improvements, so we focus on the alignment quality and try to enhance the system performance with morpho-syntactic knowledge.

In [Morrissey & Way⁺ 07], a collaborative effort with Dublin City University, we employed two different decoders but only expressed our intention to combine them. In Section 7.4.4, we will produce different translations with various techniques that perform similarly on the error measures but lead to an improvement when applied in a system combination set-up.

7.4.1 On the Choice of the Development Corpus

The purpose of a development set is to obtain scaling factors λ_m for the feature functions so that the translation system generalizes well to unseen data. It is crucial to keep the development set separate from the training and the test set, which is not a big constraint for normal-sized corpora since the withheld sentences only make up a negligible portion (usually around .1%) of the whole training set. In our case, however, holding back a development set of the same size as our test set strips away 20% of the training material. In this section, we are therefore looking for alternative optimization approaches.

First, we define a traditional split into disjoint training and development sets as our *baseline*. In [Massó & Badia 10], the authors claim that the best way to optimize the scaling factors on their corpus is to train them on the complete training set, thus not utilizing a development set at all. This approach, which we will denote as *training-on-training*, obviously bears the danger of over-fitting. Instead, we create five different translation systems,

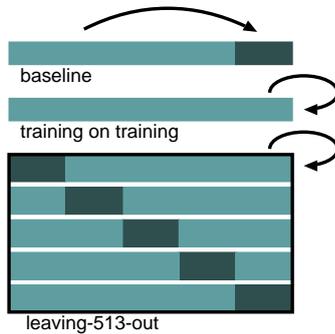


Figure 7.2: Graphical representation of the different optimization methods (Section 7.4.1)

Table 7.8: Results for the hierarchical phrase-based decoder on the various development set decisions (Section 7.4.1)

	BLEU	TER
Baseline	22.0	73.9
Training-on-training	17.7	81.6
Leaving-513-out	23.0	72.0

each trained on a disjoint sub-set of the training corpus. In each optimization iteration, we translate the parts of the training set with the same scaling factors but with different systems, concatenate the n -best lists of all individual systems for a complete training set translation, and optimize them jointly. We decided to split our training set into five disjoint sets, each excluding 513 sentences, and call this cross-validation procedure *leaving-513-out*. See Figure 7.2 for a graphical representation.

The results can be seen in Table 7.8. We conducted a similar experiment already in [Stein & Schmidt⁺ 10], where the training-on-training method was lagging behind and performed ($p < .1$) worse than the classical approach. The optimization run for training-on-training deteriorated completely: when optimizing on the (known) training material, the method obtained a BLEU score of 89.9, but was not at all able to generalize on the with-held test set. The leaving-513-out method, on the other hand, was able to obtain significantly better results on test than the classical split into training-development-test in this approach. We are thus not able to reproduce the findings of [Massó & Badia 10] on our corpus.

Table 7.9: RWTH-PHOENIX results for the phrase-based decoder using different optimization criteria (Section 7.4.2)

Criterion	Test		
	BLEU	TER	PER
BLEU	24.3	65.7	57.9
TER	22.6	60.8	55.0
BLEU - TER	22.8	60.9	54.9
WER	22.2	60.8	55.1
PER	23.0	61.4	54.9

7.4.2 On the Choice of the Training Criterion

Current SMT systems are usually optimized on BLEU, that is, the scaling factors λ_m of the log-linear model (Eqn. 3.16) are adjusted such that the BLEU score on a development corpus is maximized. [Morrissey & Way 06] however argued whether the standard metrics such as BLEU, the word error rate (WER, computed as the Levenshtein distance [Levenshtein 66]) or PER are suitable for sign languages, since they were unable to produce interpretable results. More precisely, the BLEU metric could in some instances not find a single 4-gram that was correct and thus reports an overall score of 0.

In the previous sections, we have already shown that an SMT system can be set up and trained using the standard techniques, and the authors themselves have already stated in [Morrissey & Way⁺ 07] that with newer data and better translation systems, this problem no longer exists. However, the question remains which evaluation metric is most suitable for training (for spoken languages, see [Och 03, Mauser & Hasan⁺ 08]). In this section, we optimized the phrase-based translation system on the different metrics. The results are summarized in Table 7.9.

As expected, optimizing the system on BLEU leads to optimal performance with regard to that measure, with all other systems being significantly worse ($p < .01$). The PER system ranks second according to the BLEU score, and performs much better in terms of TER. We thus conducted a human evaluation on the PER and the BLEU systems and analyzed 100 random, blind test sentences as to which translation was closer to the reference, in terms of adequacy. Only the option “better than” or “worse than” was given (while the order of the systems was blind and permuted). Evaluator A found 34 sentences indistinguishable in quality, and preferred 42 sentences from the BLEU system, compared to 24 sentences where the PER had the better translation. Evaluator B found (on a different set) 39

Table 7.10: RWTH-PHOENIX results for the hierarchical phrase-based decoder on the alignment merging strategies (Section 7.4.3)

	PBT		JANE	
	BLEU	TER	BLEU	TER
Baseline	24.6	64.9	24.3	67.3
Crunched Alignment	25.1	64.2	25.0	66.5

sentences indistinguishable, and gave favour to 30 BLEU system translations while preferring 31 sentences from the PER system. We felt that this did not justify changing the optimization measure, and, for the following experiments, we therefore stuck to BLEU as the optimization criterion.

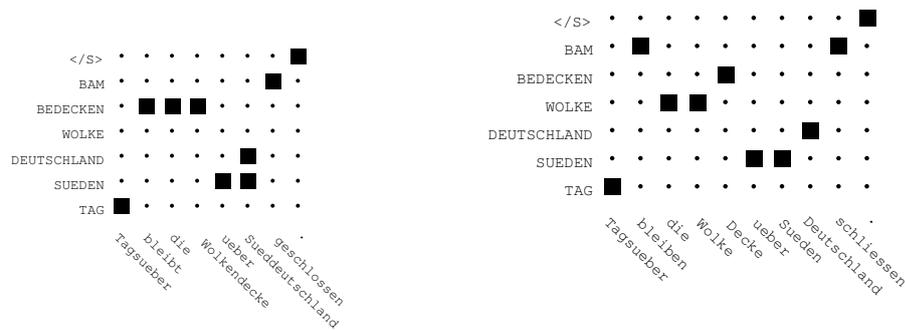
7.4.3 Linguistically Motivated Approaches

The splitting and stemming of spoken German as in Section 7.3.2 is not applicable for this translation direction, since it would now change the target language and no longer match the reference, and reverting the changes in a post-processing step is likely to introduce further errors. We can, however, improve the alignment quality with the knowledge of the compound word splittings. When splitting the words, we remember for each word whether and where it was split, train the alignments with GIZA++ on the split corpus, but crunch the alignment back to the previous positions. By doing so, the alignment indices match the original word positions, but GIZA++ should be able to make better estimates of its models. See Figure 7.3 for a graphical representation of this technique.

The results can be found in Table 7.10. We already applied this method successfully to medium-scale German corpora in [Popović & Stein⁺ 06], and for this task, both systems improve over the best system from Table 7.3, as well.

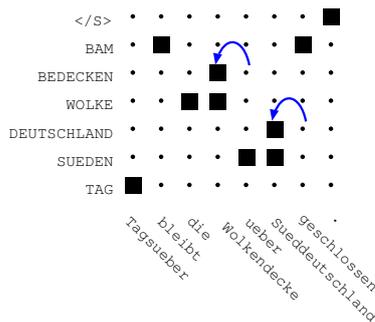
7.4.4 System Combination

Many other experiments we conducted resulted in systems with no significant improvements but only comparative results. Nevertheless, the hypotheses were distinct enough to be used in system combination. We generated alternative hypotheses with the following translation system extensions: for PBT, we use extended lexicon models, i.e. triplets and discriminative word lexica, as explained in Section 4.3.2. For Jane, we use syntactic models, i.e. soft syntactic labels and parse match as explained in Section 5.2 and Section 5.3.



(a) Baseline alignment.

(b) Split alignment.



(c) Crunched alignment.

Figure 7.3: Example of the alignment crunching effect, taken from the RWTH-PHOENIX corpus, on the sentence “Tagsüber bleibt die Wolkendecke in Süddeutschland geschlossen” (Engl.: “During the course of the day, the cloud cover in southern Germany remains dense.”). The word “Wolkendecke” (“cloud cover”) is a singleton, but “Wolke” (“cloud”) is of course well-known and “Decke” is known from “Schneedecke” (“snow cover”). Thus, in (a) the alignment has errors, but for compound split German in (b) the quality is much better. After crunching the alignment in (c), the alignment structure matches the original German sentence.

Table 7.11: RWTH-PHOENIX results for the phrase-based decoder using extended lexicon models (Section 7.4.4)

	BLEU	TER
Baseline	24.5	64.9
Triplet	24.6	66.6
DWL	24.5	65.1
DWL+Triplet	24.6	64.1

Table 7.12: RWTH-PHOENIX results for the JANE decoder using syntactic models (Section 7.4.4)

	BLEU	TER
Baseline	24.3	67.3
Syntactic Labels	24.3	67.3
Parsematch	24.5	67.0
Syntactic Labels + Parsematch	24.0	68.1

The results for extended lexicon models are summarized in Table 7.11. While the triplet model and its combination with the DWL model do not lead to improvements over the baseline in terms of BLEU, the TER score improves when applying both models on the training data, but none of the differences are statistically significant except for the bad TER score in the triplet result. It seems that in the case of small corpora such as sign language translation, the extended lexicon models tend to help less than on large corpora.

The results for the syntactic models can be found in Table 7.12. Consistent with our findings in [Stein & Schmidt⁺ 10], the translation quality does not improve over the baseline; none of the results are significant, in either direction. It seems that the methods are already too sophisticated to work properly on the small training set, and that including them in our optimization framework distracted the decoder more than it helped.

Results

See Table 7.13 for an overview of the systems that were chosen and the results of the system combination. The resulting hypothesis improves over the

Table 7.13: RWTH-PHOENIX results of the final system combination (Section 7.4.4)

System	Method	BLEU	TER
Hierarchical System	Union Alignment	24.3	67.3
	Crunched Alignment	25.0	66.5
	Syntactic labels + Parsematch	24.0	68.1
Phrase-based System	Intersection Alignment	24.6	65.1
	Crunched Alignment	25.1	64.2
	DWL + Triplet	24.6	64.1
System Combination		26.0	63.8

best single system by 0.9 BLEU absolute (3.6% rel.), statistically significant with $p < .05$, and by 0.4 TER absolute (0.6% rel.).

7.5 Conclusion

In this chapter, we focussed on the preparation and optimization of sign language machine translation.

By comparison of different decoder paradigms, we investigated preprocessing with sentence-end markers, extraction heuristics, and categorization wherever applicable. Morphologically complex spoken languages as source language should be preprocessed with proper morpho-syntactic tools in order to facilitate the translation procedure. Parallel input channels of a sign language as source language require an adequate mapping into a single stream.

The second part of this chapter introduced suitably tailored techniques for the optimization of scarce resource MT. We suggest a kind of cross-validation for the scaling factor estimation, and proposed a technique for improving the alignment that is able to work in compound split knowledge without changing the reference. Finally, we derived several translation systems for system combination.

Overall, we tried to cover some open issues that came up in the recent literature on sign language translation. It might be interesting to see whether these findings hold true for other under-resourced language pairs as well.

Scientific Achievements

In this chapter, we will conclude our work based on the scientific goals as stated in Section 2, and give an outlook towards future research directions.

- In Chapter 4, we introduced the open-source software Jane, which was developed in the course of this thesis. Jane is a state-of-the-art hierarchical toolkit made available to the scientific community. The system in its current state is stable and efficient enough to handle even large-scale tasks such as the WMT and NIST evaluations, while producing highly competitive results.
- The system implements the standard hierarchical phrase-based translation approach and different extensions that further enhance the performance of the system. Some of them, like additional reordering and lexicon models, are exclusive to Jane.
- In Section 5.2, we presented parse matching as an easy to implement syntactic enhancement. Contrary to the findings of [Chiang 05], we observed statistically significant improvement on two out of three NIST Chinese–English test sets: 0.6–0.8% BLEU absolute (1.8–2.6% rel.) and 0.6–1.1% in TER absolute (1.0–1.7% rel.). We believe that this can be attributed to a slightly more complex model that not only marks syntactically valid phrases with a binary marker feature, but also takes the distance to the next valid node into account.
- In Section 5.3, we reimplemented the soft syntactic label approach as in [Venugopal & Zollmann⁺ 09], but restricted the number of possible labels by relying on the parse match algorithm described above. The method improved the translation quality measures considerably, by up to 2.2% BLEU points absolute (6.6% rel.) in some test sets for the Chinese–English translation direction, but also by a statistically

significant 0.7% BLEU absolute (3.0% rel.) on the German–French WMT '10 test set.

- In Section 5.4, we presented our implementation decisions when extending the string-to-dependency approach as in [Shen & Xu⁺ 08], so that the phrase table is not reduced to phrases which match certain dependency conditions. By including merging errors during the dependency tree reconstruction phase, we were able to see statistically significant improvement on the NIST Chinese–English '08 test set. Despite the fact that the other test sets hardly changed at all, the dependency language model rescoring performed better if the decoder was optimized on the merging error features. With recombination, we presented statistically significant improvements of up to 1.1% BLEU absolute (3.5% rel.) on the NIST Chinese–English task. The improvement for German–French was only slight, and the effect was mostly seen in terms of a lower TER score.
- All three syntactic methods have been shown to improve the translation quality individually. In Section 5.5, we also applied all of them simultaneously and compared their individual performance as well as their combination ability on a common baseline. There seems to have been a slight saturation when using all methods combined. While for the eval set of the German–French task the BLEU score was highest, with a gain of 0.8% absolute (2.2% rel.) over the baseline, the Chinese–English system does not improve over the single syntactic labels system.
- In Chapter 6, we focussed on sign language corpora. We showed that, even in the current literature, the reading and writing skills of many Deaf show discrepancies when compared with hearing persons. We analyzed the RWTH-PHOENIX corpus based on running words and trigram perplexity, and by comparing the statistics to a small-sized spoken language corpus (IWSLT). We found the domain of weather forecasting to be suitable for our purposes. While the number of running words in RWTH-PHOENIX is merely 10% of the IWSLT data, the type-token ratio is comparable, as is the numbers of singletons.
- In the same chapter, we also introduced and analyzed a new corpus in spoken Dutch and Sign Language of the Netherlands, the Corpus-NGT. We found it to be the next step towards a broader domain SLMT system, with the recording setting allowing for more natural signing by Deaf native speakers. The Corpus-NGT is scientifically quite interesting, since it is in the challenging domain of guided discussion on Deaf issues, and since the transcription lists both hands in individual input streams.

- In Section 7.3, we established several suitably tailored techniques for the preparation of sign language data collections, to enhance MT performance. With proper preprocessing of the source sentences, e.g. by introducing sentence-end markers to the sign language, and by applying appropriate extraction heuristics, i.e. the extraction of unaligned words, we were able to statistically significant improve over the baseline by 3–4% in BLEU absolute (11.6–21.6% rel.) and 2–3% in TER absolute (3.3–5.7% rel.) on the RWTH-PHOENIX task.
- We presented some solutions when a morphologically complex spoken language is involved and an automatic parser is available. If this language is the source language, we could improve translation quality by up to 0.5% BLEU absolute (3.3% rel.) and 1.3% TER absolute (1.7% rel.) by splitting words at their strong compound word points.
- In Section 7.4, we introduced suitably tailored techniques for the optimization of scarce resource MT. We suggest a kind of cross-validation for the scaling factor estimation, and proposed a technique for improving the alignment that is able to work in compound split knowledge without changing the reference. Finally, we derived several translation systems for a system combination, which produces a statistically significant improvement of 0.9% BLEU absolute (3.6% rel.) over the best single system.

In general, this thesis focussed on feature functions that are soft in the sense that they do not restrict the decoder from certain translation possibilities. A large portion of the models covered in the previous chapters try to incorporate linguistic knowledge, but we also made use of purely statistical models such as discriminative word lexica. Even if a reduction of the phrase table size sometimes leads to improvements in translation quality (e.g. [Johnson & Martin⁺ 07]), we believe that even the most careful pruning always throws away useful information. We consider soft features to be a preferable approach, since these can be used to penalize certain phrases, but they will never narrow the number of possibilities for the decoder.

What do our findings suggest for the field of sign language machine translation? The enthusiasm over a strong baseline put aside, there are still many open problems that have to be covered in the future. The weather forecast domain might be suitable for machine translation, but the impact and usefulness for the Deaf community is rather low. The possible benefits of the domain of open discussions about Deaf related issues as found in the NGT corpus are much higher, but the influence of an erroneous sign language recognition input on the machine translation output has yet to be analyzed in full detail. Moreover, the parallel and spatial nature of sign languages has only been covered to a very limited degree in this article. Two input channels for the left and the right hand will not suffice when it

comes to body posture and facial expression information. Another upcoming and important challenge is the proper handling of classifiers (e.g. a hand movement indicating a car driving slopes up on a mountain road). We believe though that many of these problems are solvable, given that sufficient data is available to feed the corpus-based algorithms.

Overall, we covered some open issues that came up in the recent literature on sign language translation. It might be interesting to see whether these findings hold true for other under-resourced language pairs as well. Generally speaking, though, we believe that it is worthwhile to try as many techniques as possible even on these small corpora, since a lot of different approaches help to achieve a better translation quality and since the experiments run quite fast due to the limited size of the training material.



Curriculum vitæ

Stein, Daniel
geboren am 14.3.1980 in Düsseldorf
Staatsangehörigkeit: deutsch

Qualifikationen:

- 1998 Abitur
- 1999–2005 Studium der Informatik an der RWTH Aachen University
Abschluß Diplom Informatiker
- 2006 Beginn der Promotion



Overview of the Corpora

In this chapter, we list the statistics of the spoken language corpora that are used throughout this thesis. For a statistic of the signed language corpora, see Section 6.2 on page 66.

B.1 NIST Chinese–English

For the Chinese-English NIST, we use a selected subset of the available training material to arrive at a medium sized training corpus. Table B.1 shows the statistics of the data.

B.2 GALE Arabic–English

For Arabic-English, a phrase table has been produced from a parallel training corpus of 2.5M Arabic-English sentence pairs. Note that we did not run translation experiments on this task in the course of this thesis, but merely used the language pair to derive statistics of the syntactic models in Chapter 5. Table B.2 shows the statistics of the data.

B.3 QUAERO German–French

The corpus for German to French translation is taken from the QUAERO project and features european parliament speeches as well as news commentary collections. The corpus statistics can be found in Table B.3.

Table B.1: Data statistics for the preprocessed Chinese–English parallel training corpus.

		Chinese	English
Train:	Sentences	3,030,696	
	Running Words	77,456,152	81002954
	Vocabulary	83,128	213076
	Singletons	21,059	95544
nist02/04:	Sentences	2,666	
	Running Words	76,080	88,720
	Distinct Words	8,102	7,647
	OOVs (Running Words)	34	499
nist05:	Sentences	1,082	
	Running Words	32,096	34,390
	Distinct Words	5,159	4,805
	OOVs (Running Words)	12	188
nist06:	Sentences	1,664	
	Running Words	40,689	46,183
	Distinct Words	6,139	5,648
	OOVs (Running Words)	41	254
nist08:	Sentences	1,357	
	Running Words	34,463	42,281
	Distinct Words	6,209	5,606
	OOVs (Running Words)	16	231

Table B.2: Data statistics for the preprocessed Arabic–English parallel training corpus. Numbers have been replaced by a special category symbol.

		Arabic	English
Train:	Sentences	2,514,413	
	Running Words	54,324,372	55,348,390
	Vocabulary	264,528	207,780
	Singletons	115,171	91,390
Dev:	Sentences	1,797	
	Running Words	49,677	
	Distinct Words	9,274	
	OOV [%]	0.5	
Test:	Sentences	1,360	
	Running Words	45,095	
	Distinct Words	9,387	
	OOV [%]	0.4	

Table B.3: Data statistics for the preprocessed German–French parallel training corpus.

		German	French
Train:	Sentences	1,985,807	
	Running Words	47,287,523	53,056,873
	Vocabulary	196,306	145,042
	Singletons	79,563	52,692
Dev:	Sentences	2,121	
	Running Words	56,107	61,831
	Distinct Words	9,390	8,422
	OOVs (Running words)	637	831
Test:	Sentences	2,007	
	Running Words	54,034	58,744
	Distinct Words	9,200	8,300
	OOVs (Running words)	597	640



Sign Language Gloss Annotation Conventions

Signs are annotated as follows:

AAA The signs are usually written like the German target word. They are denoted in base form and in upper case. If one gloss represents multiple German words, these are separated by a hyphen.

Example: BILD, WIE-IMMER, NICHT-HABEN, VOR-3-JAHRE

A+B+C If a word is signed in finger alphabet, its letters are separated by a plus sign. When the name is not signed entirely, only the signed letters are written, and any additional information is put in brackets and marked appropriately (e.g. “mb” for “Mundbild”)

Example: J+E+M+E+N, B+A+R-(mb:barroso)

AAA+AAA Compounds that form a single entity but yet consist of two or more signs are concatenated with a plus, as well.

Example: EUROPA+PARLAMENT, DATEN+SCHUTZ+GESETZ

FOUR Numerals are written as words.

Example: VIER, NEUNZEHN+HUNDERT

AAA1,AAA2 Dialects or different signs with the same semantic are marked with a number.

Example: FRAU1, FRAU2

IX-(loc:a) The location of deictic signs within one sentence is labelled alphabetically.

negalp,neg Simple negation is marked with neg, negation via alpha rule is marked with negalp.

Example: negalp-KÖNNEN, neg-sagen

mb The mouthing which differs from the DGS-Lexem (e.g. “schlecht” is signed, but the mouth forms the English “bad”) is marked with mb.

Example: SCHLECHT-(mb:bad), KONKURRENZ-(mb:wetter)

ich-AAA-a Verbs that are flexed and thus carry subject and object information will be marked with lower case words. The subject is written in front of the verb, the object is written after the verb.

Example: ich-GEBEN-a, a-BESUCHEN-b

<ON>, **<OFF>**, **<PAUSE>** Onset and offset, i.e. when the hands of a signer are not visible, are marked in brackets, as are longer hesitations.

AAA++ Simple plural forms and verbs that signify a constant repetition are marked with a double plus.

Example: FRAGEN++, VERKAUFEN++, POSITION++

AAA AAA AAA Whenever the repetition refers to an actual number, it is not marked with a plural sign but the gloss is repeated.

Example: FRAGEN FRAGEN

poss-AAA Possessive signs are marked with the prefix “poss-”

Example: poss-MEIN, poss-SEIN

<EMP> Mimic without signing or non-verbal communication is marked separately.

Example: <EMP>-(mk:tja)

lh-AAA rh-AAA Phrases that are executed with different hands are transcribed with “lh” for left hands and “rh” for right hands.

Example: rh-BANK lh-VERSICHERUNG

List of Figures

1.1	Schematic diagram of a general communication system	2
1.2	[Vauquois 68] Pyramid diagram of translation approaches . .	3
3.1	Visualisation of an alignment, of valid and invalid lexical phrases and hierarchical phrases	18
3.2	Illustration of the log-linear translation model.	24
3.3	Calculation of intersection points in Och’s MERT algorithm. Figure (a) shows the 3-best list for one sentence and has three intersection points, of which only i' and i'' result in a change of the selected best sentence. Figure (b) shows the 2-best list of another sentence. Its intersection point will be added to the list of relevant intersection points.	26
4.1	Workflow of the extraction procedure	31
4.2	Implementation of a node in the prefix tree. The s_i denote the indexes of the successors of a node.	39
5.1	Stanford phrase structure parse for the sentence “China is becoming a country ruled by laws.”	45
5.2	Stanford dependency parse for the sentence “In recent years, the textile industry in China faced serious difficulties.” and an example for some of the resulting dependency LM probabilities.	46
5.3	All corpora: percentage of the distinct phrase table entries having a phrase match word distance of value x or lower. . .	47
5.4	Merging errors in dependency tree reconstruction. In (a), the inserted dependency structures fit into the higher structure, and no merging error occurs. In (b), neither derivation has the same direction as the higher structure, so that three dependency directions need to be adopted.	52

6.1	Two example transcriptions of signs in (a) Stokoe notation and in (b) the Hamburger Notation System.	62
6.2	Samples for admissible hand configurations and their combination in different sign languages. Figure (a) is a common hand shape that occurs frequently in DGS, whereas Figure (b) is not part of any meaningful phrase but occurs in Italian Sign Language. Likewise, the gesture in Figure (c) is an admissible combination of two hands that occurs in DGS, whereas the gesture in Figure (d) is not, although it appears in the Chinese sign language.	63
6.3	Minimal pair example in DGS. The hand situation, the place of articulation and the movement remains the same, but the hand configuration differs. Figure (a) means “SAY”, Figure (b) means “ASK”.	64
6.4	Examples for different facial expressions that change the meaning of a sign. For example, a certain gesture meaning “QUESTION” in DGS would have the facial expression as in Figure (a), but would mean “INTERROGATION” if it had the facial expression as in Figure (b).	65
6.5	Illustration of two flexed forms of the verb “GIVE”. The sign in Figure (a) is a simple forward movement and means that the signer is giving something to a single person, whereas the sign in Figure (b), sharing the same hand form but executed with a huge arc, means that the signer is giving something to a group of people.	65
6.6	Old and new television format used in the Phoenix television channel	68
6.7	Example translations taken from the RWTH-PHOENIX corpus. Sentence (a) features a rhetoric question (literal translation: “The most friendly weather remains where? In the south.”), a common grammatic device in sign languages. Sentence (b), however, is quite close to the German grammar and can be considered to be a signing glitch by the interpreter.	69
6.8	RWTH-PHOENIX: ratio of the vocabulary size compared to the running words, and ratio of the singletons compared to the vocabulary.	70
6.9	RWTH-PHOENIX: vocabulary size increase per session and notable topics of the specific broadcasts.	71
6.10	Screenshots from the Corpus-NGT corpus (Section 6.2.2)	73
6.11	Example sentences from the Corpus-NGT corpus. Each hand is annotated with a separated tier.	73
7.1	Extraction heuristics applied for initial phrases	84

7.2	Graphical representation of the different optimization methods (Section 7.4.1)	89
7.3	Example of the alignment crunching effect, taken from the RWTH-PHOENIX corpus, on the sentence “Tagsüber bleibt die Wolkendecke in Süddeutschland geschlossen” (Engl.: “During the course of the day, the cloud cover in southern Germany remains dense.”). The word “Wolkendecke” (“cloud cover”) is a singleton, but “Wolke” (“cloud”) is of course well-known and “Decke” is known from “Schneedecke” (“snow cover”). Thus, in (a) the alignment has errors, but for compound split German in (b) the quality is much better. After crunching the alignment in (c), the alignment structure matches the original German sentence.	92

List of Tables

4.1	Results for the additional reorderings on the Europarl German-English data. BLEU and TER results are in percentage. . . .	35
4.2	Results for the extended lexicon models on the French-English task. BLEU and TER results are in percentage.	37
4.3	Results for the extended lexicon models for the Arabic-English task. BLEU and TER results are in percentage.	38
4.4	Speed comparison Jane vs. Joshua, measured in translated words per second.	40
4.5	Results for Jane and Joshua in the WMT 2010 evaluation campaign.	41
5.1	All corpora: statistics for the dependency structures labelled “fixed on head” and “floating with children” (Section 5.4), based on the overall phrase table size. Note that our phrase tables are filtered to contain phrases from the translation sets only.	50
5.2	Results for the NIST Chinese–English task with all syntactic models. Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$	55
5.3	Results for the NIST Chinese-English task with the soft string-to-dependency model. Rescoring and Oracle Score on 1 k n -best lists. Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$	55
5.4	Results for the NIST Chinese–English task with syntactic model combination.	56
5.5	NIST Chinese–English: examples for translations with a higher syntactic soundness when relying on soft syntactic features.	57

5.6	Results for the QUAERO German–French task with all syntactic models.	57
5.7	Results for the QUAERO German–French task with the soft string-to-dependency model. Rescoring and Oracle Score on 1 k n -best lists.	58
5.8	QUAERO German–French: examples for translations with a higher syntactic soundness when relying on soft syntactic features.	58
5.9	Results for the QUAERO German–French task with all syntactic models.	59
5.10	QUAERO German–French: Comparison of the translation performance within the QUAERO project, for the evaluation in 2010.	59
6.1	Corpus statistics for the weather forecast corpus RWTH-PHOENIX (Section 6.2.1).	71
6.2	Corpus statistics for Corpus-NGT. The test set consists of multiple references (2–4, 2.5 on average). Here, only the statistics for the first sentence in each reference are presented.	72
6.3	Results for the sanity check (Section 6.3.1). Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$	76
7.1	RWTH-PHOENIX results for the hierarchical phrase-based decoder and the phrase-based decoder on a union alignment (Section 7.3.1). Significant improvements over the baseline are marked in magenta color if $p < .1$, and in blue color if $p < .05$	83
7.2	Precision, recall, F-Measure and alignment error rate (AER) of the different alignment merging strategies (Section 7.3.1)	84
7.3	RWTH-PHOENIX results for the hierarchical phrase-based decoder on the alignment merging strategies (Section 7.3.1)	84
7.4	RWTH-PHOENIX results for both decoders for the translation direction spoken German to German Sign Language (Section 7.3.2)	86
7.5	Dominant hand of all signers in the Corpus-NGT videos	87
7.6	Statistics for the left/right hand merging strategies of the Corpus-NGT corpus (Section 7.3.3)	87
7.7	Corpus-NGT Jane results of different strategies for using two gloss streams (Section 7.3.3)	87
7.8	Results for the hierarchical phrase-based decoder on the various development set decisions (Section 7.4.1)	89
7.9	RWTH-PHOENIX results for the phrase-based decoder using different optimization criteria (Section 7.4.2)	90

7.10	RWTH-PHOENIX results for the hierarchical phrase-based decoder on the alignment merging strategies (Section 7.4.3)	91
7.11	RWTH-PHOENIX results for the phrase-based decoder using extended lexicon models (Section 7.4.4)	93
7.12	RWTH-PHOENIX results for the JANE decoder using syntactic models (Section 7.4.4)	93
7.13	RWTH-PHOENIX results of the final system combination (Section 7.4.4)	94
B.1	Data statistics for the preprocessed Chinese–English parallel training corpus.	102
B.2	Data statistics for the preprocessed Arabic–English parallel training corpus. Numbers have been replaced by a special category symbol.	103
B.3	Data statistics for the preprocessed German–French parallel training corpus.	103

Bibliography

- [Almaghout & Jiang⁺ 10] H. Almaghout, J. Jiang, A. Way: CCG Augmented Hierarchical Phrase-based Machine Translation. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, pp. 211–218, Paris, France, Dec. 2010.
- [Almaghout & Jiang⁺ 11] H. Almaghout, J. Jiang, A. Way: CCG contextual labels in hierarchical phrase-based SMT. In *Proc. of the Annual Conference of the European Association for Machine Translation (EAMT)*, pp. 1–8, Leuven, Belgium, May 2011.
- [Baker & Padden 78] C. Baker, C.A. Padden: Focusing on the Nonmanual Components of ASL. In P. Siple, editor, *Understanding Language through Sign Language Research. (Perspectives in Neurolinguistics and Psycholinguistics)*, pp. 27–57, New York, San Francisco, London, Nov. 1978. Academic Press.
- [Battison 78] R. Battison: *Lexical Borrowing in American Sign Language*. Linstok Press, Silver Spring, MD, 1978.
- [Bauer & Kraiss 01] B. Bauer, K.F. Kraiss: Towards an Automatic Sign Language Recognition System Using Subunits. In *Gesture and Sign Language in Human-Computer Interaction. International Gesture Workshop GW 2001*, pp. 64–75, London, April 2001. Springer.
- [Becker 10] C. Becker: Lesen und Schreiben Lernen mit einer Hörschädigung. *Unterstützte Kommunikation*, Vol. 1, pp. 17–21, 2010.
- [Bellugi & Fischer 72] U. Bellugi, S. Fischer: A Comparison of Sign Language and Spoken Language. *Cognition*, Vol. 1, pp. 173–200, 1972.

- [Bentley & Ottmann 79] J.L. Bentley, T.A. Ottmann: Algorithms for Reporting and Counting Geometric Intersections. *IEEE Trans. Comput.*, Vol. 28, No. 9, pp. 643–647, 1979.
- [Bertoldi & Tiotto⁺ 10] N. Bertoldi, G. Tiotto, P. Prinetto, E. Piccolo, F. Nunnari, V. Lombardo, A. Mazzei, R. Damiano, L. Lesmo, A.D. Principe: On the Creation and the Annotation of a Large-scale Italian-LIS Parallel Corpus. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 19–22, Valletta, Malta, May 2010.
- [Birch & Blunsom⁺ 09] A. Birch, P. Blunsom, M. Osborne: A Quantitative Analysis of Reordering Phenomena. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pp. 197–205, Athens, Greece, March 2009. Association for Computational Linguistics.
- [Braem 95] P.B. Braem: *Einführung in die Gebärdensprache und ihre Erforschung*. Internationale Arbeiten zur Gebärdensprache und Kommunikation Gehörloser; Bd. 11. Signum-Verlag, Hamburg, Germany, third edition, 1995.
- [Brown & Cocke⁺ 90] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, P.S. Rossin: A Statistical Approach to Machine Translation. *Computational Linguistics*, Vol. 16, No. 2, pp. 79–85, June 1990.
- [Brown & Della Pietra⁺ 93] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, R.L. Mercer: The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, June 1993.
- [Brugman & Russel 04] H. Brugman, A. Russel: Annotating Multi-media / Multimodal Resources with ELAN. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pp. 2065–2068, Lisbon, Portugal, May 2004.
- [Bungeroth & Ney 04] J. Bungeroth, H. Ney: Statistical Sign Language Translation. In *LREC 2004, Workshop proceedings: Representation and Processing of Sign Languages*, pp. 105–108, Lisbon, Portugal, May 2004.
- [Bungeroth & Stein⁺ 06] J. Bungeroth, D. Stein, P. Dreuw, M. Zahedi, H. Ney: A German Sign Language Corpus of the Domain Weather Report. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pp. 2000–2003, Genoa, Italy, May 2006.
- [Bungeroth & Stein⁺ 08] J. Bungeroth, D. Stein, P. Dreuw, H. Ney, S. Morrissey, A. Way, L. van Zijl: The ATIS Sign Language Corpus. In *Interna-*

- tional Conference on Language Resources and Evaluation*, 4, Marrakech, Morocco, May 2008.
- [Byrd & Lu⁺ 95] R.H. Byrd, P. Lu, J. Nocedal, C. Zhu: A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, Vol. 16, No. 5, pp. 1190–1208, 1995.
- [Candito & Crabb⁺ 10] M.H. Candito, B. Crabb, P. Denis: Statistical French dependency parsing: treebank conversion and first results. In *Proceedings of LREC'2010*, pp. 1840–1847, La Valletta, Malta, May 2010.
- [Chappelier & Rajman 98] J.C. Chappelier, M. Rajman: A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of the First Workshop on Tabulation in Parsing and Deduction*, pp. 133–137, April 1998.
- [Chiang 05] D. Chiang: A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 263–270, Ann Arbor, Michigan, USA, June 2005.
- [Chiang 10] D. Chiang: Learning to Translate with Source and Target Syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1443–1452, Uppsala, Sweden, July 2010.
- [Chiang & Knight⁺ 09] D. Chiang, K. Knight, W. Wang: 11,001 New Features for Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 218–226, Boulder, Colorado, June 2009.
- [Chiu & Wu⁺ 07] Y. Chiu, C. Wu, H. Su, C. Cheng: Joint Optimization of Word Alignment and Epenthesis Generation for Chinese to Taiwanese Sign Synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 29(1), pp. 28–39, Jan. 2007.
- [Chomsky 56] N. Chomsky: Three Models for the Description of Language. *IEEE Transaction on Information Theory*, Vol. 2, No. 3, pp. 113–124, 1956.
- [Chomsky 68] N. Chomsky: Quine’s Empirical Assumptions. In *Words and Objections. Essays on the Work of W.V. Quine*, pp. 53–68, Dordrecht, Netherlands, Dec. 1968.
- [Cocke 69] J. Cocke: *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

- [Crammer & Singer 03] K. Crammer, Y. Singer: Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, Vol. 3, pp. 951–991, 2003.
- [Crasborn & Sloetjes 08] O. Crasborn, H. Sloetjes: Enhanced ELAN Functionality for Sign Language Corpora. In Crasborn, Hanke, Efthimiou, Zwitterlood, Thoutenhoofd, editors, *Construction and Exploitation of Sign Language Corpora. 3rd Workshop on the Representation and Processing of Sign Languages at LREC 2008*, pp. 39–43, Paris, France, 2008. ELDA.
- [Crasborn & van der Kooij⁺ 04] O. Crasborn, E. van der Kooij, A. Nonhebel, W. Emmerik.: *ECHO Data Set for Sign Language of the Netherlands (NGT)*. Department of Linguistics, Radboud University Nijmegen, 2004.
- [Crasborn & Zwitterlood 08] O. Crasborn, I. Zwitterlood: The Corpus NGT: An Online Corpus for Professionals and Laymen. In Crasborn, Hanke, Efthimiou, Zwitterlood, Thoutenhoofd, editors, *Construction and Exploitation of Sign Language Corpora. 3rd Workshop on the Representation and Processing of Sign Languages at LREC 2008*, pp. 44–49, Paris, France, May 2008. ELDA.
- [d’Armond L. Speers 02] d’Armond L. Speers: *Representation of American Sign Language for Machine Translation*. Ph.D. thesis, Georgetown University, Washington D.C., December 2002.
- [de Marneffe & MacCartney⁺ 06] M.C. de Marneffe, B. MacCartney, C.D. Manning: Generating Typed Dependency Parses from Phrase Structure Parses. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pp. 449–454, Genoa, Italy, May 2006.
- [DeNeefe & Knight⁺ 07] S. DeNeefe, K. Knight, W. Wang, D. Marcu: What Can Syntax-based MT Learn from Phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 755–763, Prague, Czech Republic, June 2007.
- [D’Haro & San-Segundo⁺ 08] L.F. D’Haro, R. San-Segundo, R.d. Córdoba, J. Bungeroth, D. Stein, H. Ney: Language Model Adaptation For a Speech to Sign Language Translation System Using Web Frequencies and a Map Framework. In *Interspeech 2008*, Brisbane, Australia, Sept. 2008.
- [Dreuw & Forster⁺ 10a] P. Dreuw, J. Forster, Y. Gweth, D. Stein, H. Ney, G. Martinez, J. Verges Llahi, O. Crasborn, E. Ormel, W. Du, T. Hoyoux, J. Piater, J.M. Moya Lazaro, M. Wheatley: SignSpeak – Understanding, Recognition, and Translation of Sign Languages. In *4th Workshop on*

the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, Valletta, Malta, May 2010.

- [Dreuw & Forster⁺ 10b] P. Dreuw, J. Forster, Y. Gweth, D. Stein, H. Ney, G. Martinez, J. Verges Llahi, O. Crasborn, E. Ormel, W. Du, T. Hoyoux, J. Piater, J.M. Moya Lazaro, M. Wheatley: SignSpeak – Understanding, Recognition, and Translation of Sign Languages. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Malta, May 2010.
- [Dreuw & Stein⁺ 07] P. Dreuw, D. Stein, H. Ney: Enhancing a Sign Language Translation System with Vision-Based Features. In *International Workshop on Gesture in Human-Computer Interaction and Simulation*, pp. 18–20, Lisbon, Portugal, May 2007.
- [Dreuw & Stein⁺ 08] P. Dreuw, D. Stein, T. Deselaers, D. Rybach, M. Zahedi, J. Bungeroth, H. Ney: Spoken Language Processing Techniques for Sign Language Recognition and Translation. *Technology and Disability*, Vol. 20, No. 2, pp. 121–133, June 2008.
- [Dreuw & Stein⁺ 09] P. Dreuw, D. Stein, H. Ney: Enhancing a Sign Language Translation System with Vision-Based Features. *Gesture-Based Human-Computer Interaction and Simulation*, Vol. 5085, No. 1, pp. 108–113, Jan. 2009.
- [Efthimiou & Fotinea⁺ 09] E. Efthimiou, S.E. Fotinea, C. Vogler, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, J. Segouat: Sign Language Recognition, Generation, and Modelling: A Research Effort with Applications in Deaf Communication. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction. Addressing Diversity*, Vol. 5614 of *Lecture Notes in Computer Science*, pp. 21–30. Springer Berlin / Heidelberg, 2009.
- [Fiscus 97] J.G. Fiscus: A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. of Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 347–354, 1997.
- [Fletcher & Powell 63] R. Fletcher, M.J.D. Powell: A Rapidly Convergent Descent Method for Minimization. *The Computer Journal*, Vol. 6, No. 2, pp. 163–168, 1963.
- [Forster & Stein⁺ 10] J. Forster, D. Stein, E. Ormel, O. Crasborn, H. Ney: Best Practice for Sign Language Data Collections Regarding the Needs of Data-Driven Recognition and Translation. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 92–97, Valletta, Malta, May 2010.

- [Groves & Way 05] D. Groves, A. Way: Hybrid example-based SMT: the best of both worlds? In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, ParaText '05, pp. 183–190, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [Hasan & Ganitkevitch⁺ 08] S. Hasan, J. Ganitkevitch, H. Ney, J. Andrés-Ferrer: Triplet Lexicon Models for Statistical Machine Translation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 372–381, Oct. 2008.
- [Hasan & Ney 09] S. Hasan, H. Ney: Comparison of Extended Lexicon Models in Search and Rescoring for SMT. In *Proc. of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Vol. Companion Volume: Short Papers, pp. 17–20, Boulder, CO, USA, June 2009.
- [He & Way 09] Y. He, A. Way: Improving the objective function in minimum error rate training. In *Proc. of the Machine Translation Summit*, Ottawa, Canada, Aug. 2009.
- [Heger & Wuebker⁺ 10] C. Heger, J. Wuebker, M. Huck, G. Leusch, S. Mansour, D. Stein, H. Ney: The RWTH Aachen Machine Translation System for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pp. 93–97, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
- [Hermans & Knoors⁺ 08a] D. Hermans, H. Knoors, E. Ormel, L. Verhoeven: Modeling Reading Vocabulary Learning in Deaf Children in Bilingual Education Programs. *Journal of Deaf Studies and Deaf Education*, Vol. 13:2, pp. 155–174, Spring 2008.
- [Hermans & Knoors⁺ 08b] D. Hermans, H. Knoors, E. Ormel, L. Verhoeven: The Relationship Between the Reading and Signing Skills of Deaf Children in Bilingual Education Programs. *Journal of Deaf Studies and Deaf Education*, Vol. 13:4, pp. 519–530, Fall 2008.
- [Huang & Chiang 07] L. Huang, D. Chiang: Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 144–151, Prague, Czech Republic, June 2007.
- [Huck & Vilar⁺ 11a] M. Huck, D. Vilar, D. Stein, H. Ney: Advancements in Arabic-to-English Hierarchical Machine Translation. In *15th Annual Conference of the European Association for Machine Translation*, pp. 273–280, Leuven, Belgium, May 2011. European Association for Machine Translation.

- [Huck & Vilar⁺ 11b] M. Huck, D. Vilar, D. Stein, H. Ney: Lightly-Supervised Training for Hierarchical Phrase-Based Machine Translation. In *EMNLP 2011 Workshop on Unsupervised Learning in NLP*, pp. 91–96, Edinburgh, UK, July 2011.
- [Huck & Wuebker⁺ 11] M. Huck, J. Wuebker, C. Schmidt, M. Freitag, S. Peitz, D. Stein, A. Dagnelies, S. Mansour, G. Leusch, H. Ney: The RWTH Aachen Machine Translation System for WMT 2011. In *EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pp. 405–412, Edinburgh, UK, July 2011.
- [Huenerfauth 03] M. Huenerfauth: A Survey and Critique of American Sign Language Natural Language Generation and Machine Translation Systems. Technical Report MS-CIS-03-32, University of Pennsylvania, September 2003.
- [Johnson & Martin⁺ 07] J.H. Johnson, J. Martin, G. Foster, R. Kuhn: Improving Translation Quality by Discarding Most of the Phrasetable. In *Proceedings of EMNLP-CoNLL*, pp. 967–975, Prague, Czech, 2007.
- [Kanis & Müller 09] J. Kanis, L. Müller: Advances in Czech – Signed Speech Translation. In *Lecture Notes in Computer Science*, Vol. 5729, pp. 48–55. Springer, 2009.
- [Kanis & Zahradil⁺ 06] J. Kanis, J. Zahradil, F. Jurčiček, L. Müller: Czech-sign Speech Corpus for Semantic Based Machine Translation. *Lecture Notes in Artificial Intelligence*, Vol. 4188, pp. 613–620, 2006.
- [Kasami 65] T. Kasami: An Efficient Recognition and Syntax Analysis Algorithm for Context-Free Languages. Technical report, Hawaii University Honolulu Department of Electrical Engineering, July 1965.
- [Klein & Manning 03] D. Klein, C.D. Manning: Accurate Unlexicalized Parsing. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 423–430, Sapporo, Japan, July 2003.
- [Klima & Bellugi 79] E.S. Klima, U. Bellugi: *The Signs of Language*. Harvard University Press, Cambridge, UK, 1979.
- [Koehn 04] P. Koehn: Statistical Significance Tests for Machine Translation Evaluation. In D. Lin, D. Wu, editors, *Proc. of EMNLP 2004*, Barcelona, Spain, July 2004.
- [Koehn & Arun⁺ 08] P. Koehn, A. Arun, H. Hoang: Towards better machine translation quality for the German–English language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pp. 139–142, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

- [Koehn & Hoang⁺ 07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst: Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 177–180, Prague, Czech Republic, June 2007.
- [Koehn & Och⁺ 03] P. Koehn, F.J. Och, D. Marcu: Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology, North American Chapter of the Association for Computational Linguistics*, pp. 54–60, Edmonton, Canada, May 2003.
- [Kramer 07] F. Kramer: *Kulturfaire Berufseignungsdiagnostik bei Gehörlosen und daraus abgeleitete Untersuchungen zu den Unterschieden der Rechenfertigkeiten bei Gehörlosen und Hörenden*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, June 2007.
- [Lambert & Banchs 06] P. Lambert, R.E. Banchs: Tuning Machine Translation Parameters with SPSA. In *Proc. of the International Workshop on Spoken Language Technology (IWSLT)*, pp. 1–7, Kyoto, Japan, Nov. 2006.
- [Levenshtein 66] V.I. Levenshtein: Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, Vol. 10, pp. 707–710, February 1966.
- [Lewis II & Stearns 68] P.M. Lewis II, R.E. Stearns: Syntax-Directed Transduction. *Journal of the ACM*, Vol. 15, No. 3, pp. 465–488, July 1968.
- [Li & Callison-Burch⁺ 09] Z. Li, C. Callison-Burch, C. Dyer, S. Khudanpur, L. Schwartz, W. Thornton, J. Weese, O. Zaidan: Joshua: An Open Source Toolkit for Parsing-Based Machine Translation. In *Proc. of the Workshop on Statistical Machine Translation*, pp. 135–139, Athens, Greece, March 2009.
- [Marton & Resnik 08] Y. Marton, P. Resnik: Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1003–1011, Columbus, Ohio, June 2008.
- [Massó & Badia 10] G. Massó, T. Badia: Dealing with Sign Language Morphemes for Statistical Machine Translation. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 154–157, Valletta, Malta, May 2010.

- [Matusov & Leusch⁺ 08] E. Matusov, G. Leusch, R.E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y.S. Lee, J.B. Marino, M. Paulik, S. Roukos, H. Schwenk, H. Ney: System Combination for Machine Translation of Spoken and Written Language. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 7, pp. 1222–1237, Sept. 2008.
- [Mauser & Hasan⁺ 08] A. Mauser, S. Hasan, H. Ney: Automatic Evaluation Measures for Statistical Machine Translation System Optimization. In *International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May 2008.
- [Mauser & Hasan⁺ 09] A. Mauser, S. Hasan, H. Ney: Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 210–218, Singapore, Aug. 2009.
- [Moore & Quirk 08] R.C. Moore, C. Quirk: Random Restarts in Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of the International Conference on Computational Linguistics (Coling)*, pp. 585–592, Manchester, UK, Aug. 2008.
- [Morrissey 08] S. Morrissey: *Data-Driven Machine Translation for Sign Languages*. Ph.D. thesis, School of Computing, Dublin City University, Dublin City University, Ireland, 2008.
- [Morrissey & Way 06] S. Morrissey, A. Way: Lost in Translation: the Problems of Using Mainstream MT Evaluation Metrics for Sign Language Translation. In *Proceedings of the 5th SALT MIL Workshop on Minority Languages at LREC'06*, pp. 91–98, Genoa, Italy, May 2006.
- [Morrissey & Way⁺ 07] S. Morrissey, A. Way, D. Stein, J. Bungeroth, H. Ney: Towards a Hybrid Data-Driven MT System for Sign Languages. In *Machine Translation Summit*, pp. 329–335, Copenhagen, Denmark, Sept. 2007.
- [Nelder & Mead 65] J. Nelder, R. Mead: The Downhill Simplex Method. *Computer Journal*, Vol. 7, pp. 308, 1965.
- [Ney 03] H. Ney: Maschinelle Sprachverarbeitung – Der statistische Ansatz bei Spracherkennung und Sprachübersetzung. *Informatik-Spektrum*, Vol. 26:2, pp. 94–102, April 2003.
- [Ney & Essen⁺ 94] H. Ney, U. Essen, R. Kneser: On Structuring Probabilistic Dependencies in Language Modelling. *Computer Speech and Language*, Vol. 8, pp. 1–38, 1994.

- [Och 03] F.J. Och: Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 160–167, Sapporo, Japan, July 2003.
- [Och & Ney 00] F.J. Och, H. Ney: A Comparison of Alignment Models for Statistical Machine Translation. In *International Conference on Computational Linguistics*, pp. 1086–1090, Saarbrücken, Germany, July 2000.
- [Och & Ney 02] F.J. Och, H. Ney: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, Pennsylvania, USA, July 2002.
- [Och & Ney 03a] F.J. Och, H. Ney: A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.
- [Och & Ney 03b] F.J. Och, H. Ney: A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, March 2003.
- [Och & Tillmann⁺ 99] F. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, Maryland, USA, June 1999.
- [Ormel & Crasborn⁺ 10] E. Ormel, O. Crasborn, E.v.d. Kooij, L.v. Dijken, E.Y. Nauta, J. Forster, D. Stein: Glossing a Multi-purpose Sign Language Corpus. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 186–191, Valletta, Malta, May 2010.
- [Papineni & Roukos⁺ 02] K. Papineni, S. Roukos, T. Ward, W.J. Zhu: Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002.
- [Peter & Huck⁺ 11] J.T. Peter, M. Huck, H. Ney, D. Stein: Soft String-to-Dependency Hierarchical Machine Translation. In *International Workshop on Spoken Language Translation*, pp. 246–253, San Francisco, California, USA, Dec. 2011.
- [Petrov & Barrett⁺ 06] S. Petrov, L. Barrett, R. Thibaux, D. Klein: Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 433–440, Sydney, Australia, July 2006.

- [Popović & Stein⁺ 06] M. Popović, D. Stein, H. Ney: Statistical Machine Translation of German Compound Words. In *5th International Conference on NLP, FinTal*, pp. 616–624, Turku, Finland, May 2006.
- [Popović & Vilar⁺ 09] M. Popović, D. Vilar, D. Stein, E. Matusov, H. Ney: The RWTH Machine Translation System for WMT 2009. In *Fourth EACL Workshop on Statistical Machine Translation*, pp. 66–69, Athens, Greece, March 2009.
- [Prillwitz 89] S. Prillwitz: *HamNoSys. Version 2.0; Hamburg Notation System for Sign Language. An Introductory Guide*. Signum Verlag, Hamburg, Germany, 1989.
- [Rexroat 97] N. Rexroat: The Colonization of the Deaf Community. *Social Work Perspectives*, Vol. 7 (1), pp. 18–26, 1997.
- [Sáfár & Marshall 01] É. Sáfár, I. Marshall: The Architecture of an English-Text-to-Sign-Languages Translation System. In G.A. et al, editor, *Recent Advances in Natural Language Processing (RANLP)*, pp. 223–228, Tzigov Chark, Bulgaria, September 2001.
- [Sandler 99] W. Sandler: Prosody in Two Natural Language Modalities. *Language and Speech*, Vol. 42, pp. 127–142, 1999.
- [Schwartz 10] L. Schwartz: Reproducible Results in Parsing-Based Machine Translation: The JHU Shared Task Submission. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics-MATR*, pp. 177–182, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
- [Shannon 48] C.N. Shannon: A Mathematical Theory of Communication. *The Bell System Technical Journal*, Vol. 27, pp. 623–656, Oct. 1948.
- [Shen & Xu⁺ 08] L. Shen, J. Xu, R. Weischedel: A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 577–585, Columbus, Ohio, June 2008.
- [Snover & Dorr⁺ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul: A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pp. 223–231, Cambridge, Massachusetts, USA, August 2006.
- [Spall & Member 92] J.C. Spall, S. Member: Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, Vol. 37, pp. 332–341, 1992.

- [Stein & Bungeroth⁺ 06] D. Stein, J. Bungeroth, H. Ney: Morpho-Syntax Based Statistical Methods for Sign Language Translation. In *Conference of the European Association for Machine Translation*, pp. 169–177, Oslo, Norway, June 2006.
- [Stein & Dreuw⁺ 07] D. Stein, P. Dreuw, H. Ney, S. Morrissey, A. Way: Hand in Hand: Automatic Sign Language to Speech Translation. In *11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pp. 214–220, Skövde, Sweden, Sept. 2007.
- [Stein & Forster⁺ 10] D. Stein, J. Forster, U. Zelle, P. Dreuw, H. Ney: Analysis of the German Sign Language Weather Forecast Corpus. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pp. 225–230, Valletta, Malta, May 2010.
- [Stein & Peitz⁺ 10] D. Stein, S. Peitz, D. Vilar, H. Ney: A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation. In *Conference of the Association for Machine Translation in the Americas 2010*, 9, Denver, USA, Oct. 2010.
- [Stein & Schmidt⁺ 10] D. Stein, C. Schmidt, H. Ney: Sign Language Machine Translation Overkill. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, pp. 337–334, Paris, France, Dec. 2010.
- [Stein & Schmidt⁺ 12] D. Stein, C. Schmidt, H. Ney: Analysis, Preparation, and Optimization of Statistical Sign Language Machine Translation. *Machine Translation*, Vol., 2012. accepted.
- [Stein & Vilar⁺ 11a] D. Stein, D. Vilar, H. Ney: A Guide to Jane, an Open Source Hierarchical Translation Toolkit. *The Prague Bulletin of Mathematical Linguistics*, Vol., No. 95, pp. 5–18, April 2011.
- [Stein & Vilar⁺ 11b] D. Stein, D. Vilar, H. Ney: *Soft Syntax Features and Other Extensions for Hierarchical SMT*, chapter 2. Springer Verlag, Jan. 2011. Accepted for publication.
- [Stokoe 60] W. Stokoe: Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in Linguistics, Occasional Papers*, Vol. 8, pp. 1–78, 1960.
- [Stolcke 02] A. Stolcke: SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pp. 901–904. ISCA, Sept. 2002.
- [Talbot & Osborne 07] D. Talbot, M. Osborne: Smoothed Bloom filter language models: Tera-scale LMs on the cheap. In *Proceedings of the 2007*

Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 468–476, Prague, Czech Republic, June 2007.

- [Vauquois 68] B. Vauquois: A survey of formal grammars and algorithms for recognition and transformation in machine translation. In *International Federation for Information Processing Congress*, Vol. 2, pp. 254–260, Edinburgh, UK, Aug. 1968.
- [Veale & Conway⁺ 98] T. Veale, A. Conway, B. Collins: The Challenges of Cross-Modal Translation: English to Sign Language Translation in the ZARDOZ System. *Journal of Machine Translation*, Vol. 13, No. 1, pp. 81–106, 1998.
- [Venugopal & Zollmann⁺ 09] A. Venugopal, A. Zollmann, N. Smith, S. Vogel: Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 236–244, Boulder, Colorado, USA, June 2009.
- [Vilar 12] D. Vilar: *Investigations on Hierarchical Phrase-based Machine Translation*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, 2012. to appear.
- [Vilar & Ney 09] D. Vilar, H. Ney: On LM Heuristics for the Cube Growing Algorithm. In *Proc. of the Annual Conference of the European Association for Machine Translation (EAMT)*, pp. 242–249, Barcelona, Spain, May 2009.
- [Vilar & Stein⁺ 08a] D. Vilar, D. Stein, H. Ney: Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, pp. 190–197, Waikiki, Hawaii, Oct. 2008.
- [Vilar & Stein⁺ 08b] D. Vilar, D. Stein, Y. Zhang, E. Matusov, A. Mauser, O. Bender, S. Mansour, H. Ney: The RWTH Machine Translation System for IWSLT 2008. In *International Workshop on Spoken Language Translation*, pp. 108–115, Honolulu, Hawaii, Oct. 2008.
- [Vilar & Stein⁺ 10a] D. Vilar, D. Stein, M. Huck, H. Ney: Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pp. 262–270, Uppsala, Sweden, July 2010.
- [Vilar & Stein⁺ 10b] D. Vilar, D. Stein, S. Peitz, H. Ney: If I Only Had a Parser: Poor Man’s Syntax for Hierarchical Machine Translation. In

- Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, 2010. Submitted, currently in review.
- [Vilar & Stein⁺ 12] D. Vilar, D. Stein, M. Huck, H. Ney: Jane: An Advanced Freely-Available Hierarchical Machine Translation Toolkit. *Machine Translation*, Vol., 2012. accepted.
- [Vogel & Ney⁺ 96] S. Vogel, H. Ney, C. Tillmann: HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th conference on Computational linguistics*, Vol. 2, pp. 836–841, Copenhagen, Denmark, Aug. 1996.
- [Watanabe & Suzuki⁺ 07] T. Watanabe, J. Suzuki, H. Tsukada, H. Isozaki: Online Large-Margin Training for Statistical Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 764–773, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- [Wauters & van Bon⁺ 06] L.N. Wauters, W.H.J. van Bon, A.E.J.M. Tellings: Reading comprehension of Dutch deaf children. *Reading and Writing: An Interdisciplinary Journal*, Vol. 19, pp. 49–76, 2006.
- [Weaver 55] W. Weaver: Translation. In *Machine Translation of Languages: Fourteen Essays*, pp. 15–23, MIT, Cambridge, Mass., USA, 1955.
- [Wrobel 01] U.R. Wrobel: Referenz in Gebärdensprachen: Raum und Person. *Forschungsberichte des Instituts für Phonetik und Sprachliche Kommunikation der Universität München*, Vol. 37, pp. 25–50, 2001.
- [Wu 97] D. Wu: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, Vol. 23, No. 3, pp. 377–403, 1997.
- [Yamada & Knight 01] K. Yamada, K. Knight: A Syntax-Based Statistical Translation Model. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 523–530, Toulouse, France, July 2001.
- [Younger 67] D.H. Younger: Recognition and Parsing of Context-Free Languages in Time n^3 . *Information and Control*, Vol. 2, No. 10, pp. 189–208, 1967.
- [Zens & Ney 06] R. Zens, H. Ney: Discriminative Reordering Models for Statistical Machine Translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting (HLT-NAACL), Workshop on Statistical Machine Translation*, pp. 55–63, New York City, June 2006.

- [Zens & Ney 07a] R. Zens, H. Ney: Efficient Phrase-Table Representation for Machine Translation with Applications to Online MT and Speech Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 492–499, Rochester, New York, April 2007.
- [Zens & Ney 07b] R. Zens, H. Ney: Efficient Phrase-table Representation for Machine Translation with Applications to Online MT and Speech Translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting*, pp. 492–499, Rochester, NY, April 2007.
- [Zens & Ney 08] R. Zens, H. Ney: Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation. In *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, pp. 195–205, Honolulu, Hawaii, Oct. 2008.
- [Zens & Och⁺ 02] R. Zens, F.J. Och, H. Ney: Phrase-Based Statistical Machine Translation. In *German Conference on Artificial Intelligence*, pp. 18–32, Aachen, Germany, Sept. 2002.
- [Zielinski & Simon 08] A. Zielinski, C. Simon: Morphisto: An Open-Source Morphological Analyzer for German. In *Proc. of the International Workshop on Finite-State Methods and Natural Language Processing*, pp. 177–184, Ispra, Italy, Sept. 2008.
- [Zollmann & Venugopal 06] A. Zollmann, A. Venugopal: Syntax Augmented Machine Translation via Chart Parsing. In *Proc. of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp. 138–141, New York, June 2006.