Universiteit van Amsterdam
Faculty of Science

Master Thesis

EBMT Based upon Two-Dimensional Alignment

Andrea Schuch

July 21, 2010

Supervisors: Prof. Dr. Remko Scha and Anouk Perquin

Abstract

Alignment models have been more prominent in the statistical (SMT) rather than the example-based machine translation (EBMT) research tradition. Word alignment is one of the oldest concepts in machine translation, and it is now readily available thanks to statistical tools. However, since real translations are rarely word-by-word, word alignment usually makes use of two linguistically unreasonable concepts: empty cepts and distortion. In this thesis, we develop a two-phase EBMT approach on the basis of a two-dimensional word alignment model. This approach avoids alignment to the empty cept and does not make use of distortion.

In two-phase (or precompiled) EBMT, translation examples are converted into translation rules during the preprocessing phase. However, since the sentence to be translated is not known at this stage, preprocessing must be *sensitive*, as it entails a great risk of losing valuable information. In order to enable a more informed matching, translation rules must remain *representative* of the original example translation. Such a translation rule we call *translation frame*, and its major task is to capture the structural discrepancies in the sentence pair. We propose to generate translation frames on the basis of a *two-dimensional alignment model*. In addition to the usual word translations – which we call *inter-sentential* dependencies – this alignment model contains *intra-sentential* dependencies, which model relations between words *within* the source and target sentences. In addition to the usual *direct* alignment between two words, this also enables *indirect* alignment of *untranslatable* words via an intra-sentential dependency to a directly aligned word. Discontiguous phrases are thus aligned in two steps: First, we align their translatable words, and then we associate their untranslatable words. Our translation frame generation method includes all words in the translation frame that take part in indirect alignment.

We implement a prototype system, which only relies on resources that are very easy to obtain: intra-sentential dependencies are computed from correlations between words, and inter-sentential dependencies are established by using single-word translations. We detailedly show how the prototype can deal with a real-corpus example, and compare its performance to two EBMT approaches of comparable simplicity (runtime and compiled). While in these experiments the protype's performance is at the same level as the baseline system's, our approach has the following advantages: The two-dimensional alignment model can exclusively rely on relationships between single words, in particular single-word translations, which are easier obtained than phrase translations. At the same time, it does not enforce translatability for every word in the sentence, but offers a treatment for untranslatable words that does not align them to the empty cept. Thirdly, intra- and inter-sentential dependencies can be computed independently of one another. Finally, alignments can be computed without distortion models or restrictions on word order, because EBMT needs them exclusively for the analysis of existing translation examples, not for generating new translations.

Acknowledgments

I would like to thank my supervisors Remko Scha and Anouk Perquin for their support and encouragement from beginning to end of this thesis. Thank you for being interested in linguistics and applications alike – for that is exactly what I had been searching for. I am also much indebted to all people – in particular Rolf Schmidt and my parents – who helped me to finish this project, because they would not give up on me. My studies have been supported by a DAAD scholarship. Afterwards, my employer Elektrobit Automotive GmbH enabled the completion of this thesis through flexible working hours.

Contents

1 Corpus-Based Approaches to Machine Translation **7**
- 1.1 Statistical Machine Translation (SMT) . . . 10
  - 1.1.1 Word-based SMT . . . 13
  - 1.1.2 Phrase-based SMT . . . 15
  - 1.1.3 Syntax-based SMT . . . 19
- 1.2 Example-Based Machine Translation (EBMT) . . . 23
  - 1.2.1 "Pure" EBMT . . . 29
  - 1.2.2 Two-Phase EBMT . . . 32
- 1.3 Comparing SMT and EBMT . . . 36
  - 1.3.1 Distinction . . . 36
  - 1.3.2 Performance . . . 39

2 An EBMT System **41**
- 2.1 Implementation . . . 41
  - 2.1.1 Lexicon . . . 42
  - 2.1.2 Runtime EBMT . . . 44
  - 2.1.3 Compiled EBMT . . . 52
- 2.2 Evaluation . . . 55
  - 2.2.1 Qualitative Evaluation . . . 55
  - 2.2.2 Quantitative Evaluation . . . 61

3 Translation Frame Generation **65**
- 3.1 Two-Dimensional Alignment . . . 66
- 3.2 Translation Frames . . . 75
- 3.3 A Prototype Implementation . . . 82
  - 3.3.1 Calculating Intra- and Inter-Sentential Dependencies . . . 82
  - 3.3.2 Translation Frame Generation Algorithm . . . 83
  - 3.3.3 A Real Corpus Example . . . 85

4 Summary and Conclusions **91**

1 Corpus-Based Approaches to Machine Translation

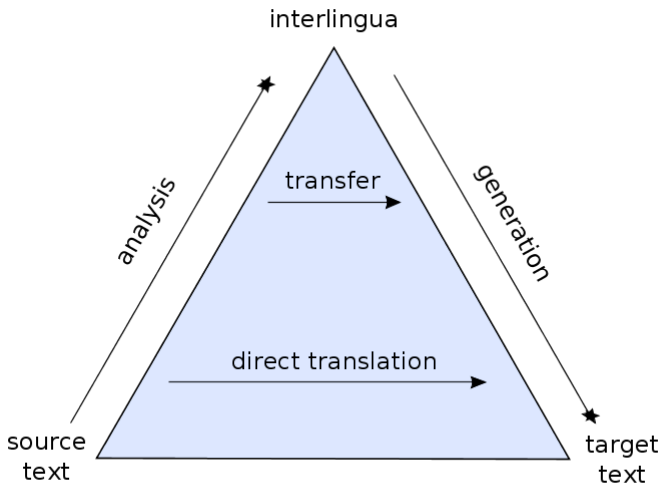

Figure 1.1: The "translation pyramid" for RBMT.

For decades, machine translation (MT) has been dominated by the so-called *rule-based* approach (RBMT), which relies on a (rich) translation lexicon and hand-crafted transfer rules. As shown in Figure 1.1, one distinguishes between three variations of RBMT (Collins, 1998):

Direct translation: Transfer rules operate directly on the source and target sentence. Relying heavily on the lexicon, this approach mostly performs word-for-word or phrase-to-phrase and is thus very restricted in its ability to cover structural discrepancies between source and target sentence.

Translation based on transfer: The transfer rules operate at one level of abstraction, translating an intermediate representation of the source sentence to an intermediate representation of the target sentence. Such representations typically contain syntactic and semantic information about the sentence.

The interlingua approach: The interlingua is a universal representation of sentence meanings, such that it acts as a mediator between source and target language, making transfer superfluous.

While the direct approach has only a limited ability, the transfer approach requires a full linguistic analysis of source and target sentence, and the interlingua approach even requires a language-independent sentence representation. In other words:

> "RBMT systems have been biased toward syntactic, semantic, and contextual analysis, which consumes considerable computing time. However, such deep analysis is not always necessary or useful for translation." (Sumita & Iida, 1991)

Obviously, "formulating linguistic rules for RBMT is a difficult job and requires a linguistically trained staff" (Sumita & Iida, 1991). However, the main problem of the RBMT approach is a *knowledge-acquisition bottleneck*:

> "the team of developers first has to fully understand the problem before it can be described in terms of rules (or their exceptions)" (Carl & Way, 2003).

According to Sumita and Iida (1991) it can be questioned if linguistics (ever) deals with all phenomena encountered in real text. Indeed, there is still a number of complex problems in MT that are "not (yet) sufficiently understood", or require "a full semantic and pragmatic analysis of the corpus, which is rarely available" (Carl & Way, 2003). As a result, RBMT systems "have a kind of inherent contradiction in themselves" (Nagao, 1984):

> "The more elaborate the RBMT becomes, the less expandable it is. Considerably complex rules concerning semantics, context, and the real world, are required in machine translation. This is the notorious AI bottleneck: not only is it difficult to add a new rule to the database of rules that are mutually dependent, but it is also difficult to build such a rule database itself. Moreover, computation using this huge and complex rule database is so slow that it forces a developer to abandon efforts to improve the system. RBMT is not easily upgraded." (Sumita & Iida, 1991)

At the same time, it has been questioned if RBMT is the best approach to translation, because

> "man does not translate simple sentences by doing a deep linguistic analysis" (Nagao, 1984).

Fortunately, the availability of computational power, electronic parallel corpora, machine-readable bilingual dictionaries, and statistical algorithms has opened the door to a new approach: *Corpus-based* (or data-driven) machine translation. In this approach,

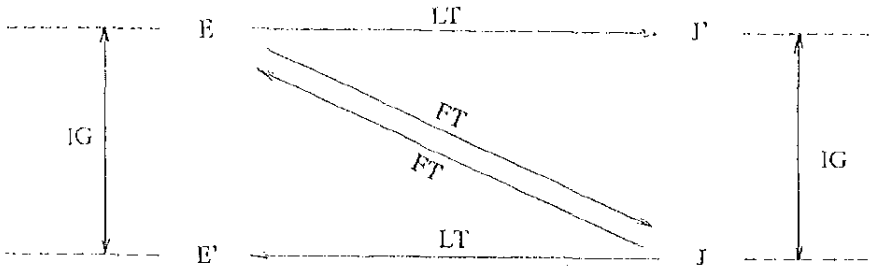

Figure 1.2: Illustration of the idiosyncratic gap (IG) (Nitta, 1986): The semantic difference between a sentence and its *literal translation (LT)*. In contrast to a *free translation (FT)*, a literal translation preserves "the wording, phrasing and various sentential structures of the original (source) sentence as much as possible" (Nitta, 1986). According to Nitta (1986) the idiosyncratic gap between two languages, e.g. English (E) and Japanese (J), should be "examined" by comparing the free translations's literal (back) translation with the original sentence.

the bottleneck is *avoided* by having translation knowledge extracted automatically from translation corpora, rather than crafted by humans. While RBMT is still widely used in commercial applications (Koehn, 2004; Ziegler, 2008), "most MT research being undertaken today is corpus-based" (Way & Gough, 2005).

Beyond pure translation memories (TM), corpus-based MT does not merely locate relevant examples but – based on these – attempts a fully automatic translation (Somers, 1999). Thus, the task of corpus-based MT is not to simply 'remember' known solutions of former problems but to create new solutions to new problems (Carl, 1998). A particular challenge is constituted by *structural discrepancies* between source and target language, which together make up the *idiosyncratic gap* between the two languages (see Figure 1.2). Some of these discrepancies simply stem from the fact that the same linguistic categories are realised differently in different languages. As an example, consider the difference in word-order between SVO-languages (English, Chinese) and SOV-languages (Japanese, Turkish), or prepositions and postpositions, etc. Further, there are (systematic) divergences inherent in the languages, many of which are described by Dorr (1994). Finally, (less systematic) differences stem from rewording, clause-reordering, etc. The less closely source and target language are related, the wider the idiosyncratic gap between them. In particular, rephrasing phenomena will be more frequent, as a literal translation might distort the original meaning of the sentence. In addition, corpus-based MT must work with *real* translation data, which may contain a freer translation even if a literal variation would also be valid.

Corpus based MT has been launched in two different approaches, *Example-based*

Figure 1.3: Illustration of Shannon's noisy channel and its application to translation from French to English: The French source sentence f is regarded as a corrupted observation of the English sentence e. The task of translating a sentence f consists in reversing the channel operations, which according to the model have caused the corruption of e (decoding). The channel operations of Brown et al. (1990)'s word-based SMT approach are: word movement, word duplication and word translation.

machine translation (EBMT) and *Statistical* machine translation (SMT). Since it is widely believed that the future of MT lies in a synthesis of both approaches (Groves & Way, 2005), we present them in turn: SMT in Section 1.1 and EBMT in Section 1.2.

1.1 Statistical Machine Translation (SMT)

SMT is the most "dominant" paradigm in today's research (Way & Gough, 2005). It applies Shannon's *noisy channel* to the problem of machine translation. As Weaver (1949) puts it:

> "A book written in Chinese is simply a book written in English which was coded into the 'Chinese code' " (Weaver, 1949)

Brown et al. (1990) provide a statistical formalisation:

> "Given a sentence T in the target language, we seek the sentence S from which the translator produced T".(Brown et al., 1990)

Note that the noisy channel view results in a switch of source and target language, which we will indicate by a subscript "channel". More specifically, Brown et al. (1990) seek the sentence $\hat{S}_{channel}$ that maximises the conditional probability $P(S_{channel}|T_{channel})$. In line with the view of Weaver (1949), this process is called *decoding*. As by the Bayes theorem $P(S|T)$ is equivalent to $\frac{P(S)P(T|S)}{P(T)}$:

$$\hat{S}_{channel} = \arg\max_{S_{channel}} P(S_{channel}|T_{channel})$$

$$= \arg\max_{S_{channel}} \frac{P(S_{channel})P(T_{channel}|S_{channel})}{P(T_{channel})}$$

Due to the maximisation the denominator can be dropped, so Brown et al. (1990) derive the following *decoder equation*:

$$\hat{S}_{channel} = \arg \max_{S_{channel}} P(S_{channel})P(T_{channel}|S_{channel})$$

The resulting equation is viewed as divided into

1. a language model $P(S_{channel})$ and
2. a translation model $P(T_{channel}|S_{channel})$.

The SMT view of the translation pyramid is shown in Figure 1.4. Switching back from the channel view to a translator's view, $P(S_{channel})$ is actually a *target language model*. To clarify, the decoder equation for translating *from French (f) to English (e)* is:

$$\hat{e} = argmax_e P(e|f) = argmax_e P(e)P(f|e)$$

The advantage of modelling the translation process indirectly via the Bayes rule, rather than estimating $P(e|f)$ directly, is that it causes the decoder equation to depend on a target language model $P(e)$, which "concentrate[s] its probability as much as possible on well-formed English strings" (Brown, Cocke, Pietra, Pietra, & Mercer, 1993).[1] Brown et al. (1993) describe how the translation model and the language model "cooperate":

> "The translation model probability is large for English strings, whether well- or ill-formed, that have the necessary words in them in roughly the right places to explain the French. The language model probability is large for well-formed English strings regardless of their connection to the French. Together, they produce a large probability for well-formed English strings that account well for the French. We cannot achieve this simply by reversing our translation models."

However, since there is no way to estimate $P(e)$ or $P(f|e)$ directly, it is necessary to *construct an approximation* of their distributions. According to Brown et al. (1993), this is "the essential question for statistical translation" – and it turned out it has remained so until today.

The target language model $P(e)$ is typically estimated by means of an n-gram language model. For example, Brown et al. (1990) use a second order Markov model, "bucketing" the words in the sentence into groups of 3:

$$P(w_1, ..., w_n) = P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P(w_i|w_{i-1}, w_{i-2})$$

Since, according to the Markov assumption, the probability of a word e_i is independent of all words preceding it at a distance greater than 3, without additional grammatical information the model will not capture long-distance dependencies.

[1] This approach has been largely adopted in SMT. A notable exception is the *alignment template model* by Och and Ney (2004), who use a log-linear approach that is "a generalisation" of the noisy channel approach, but thereby "easier to extend".

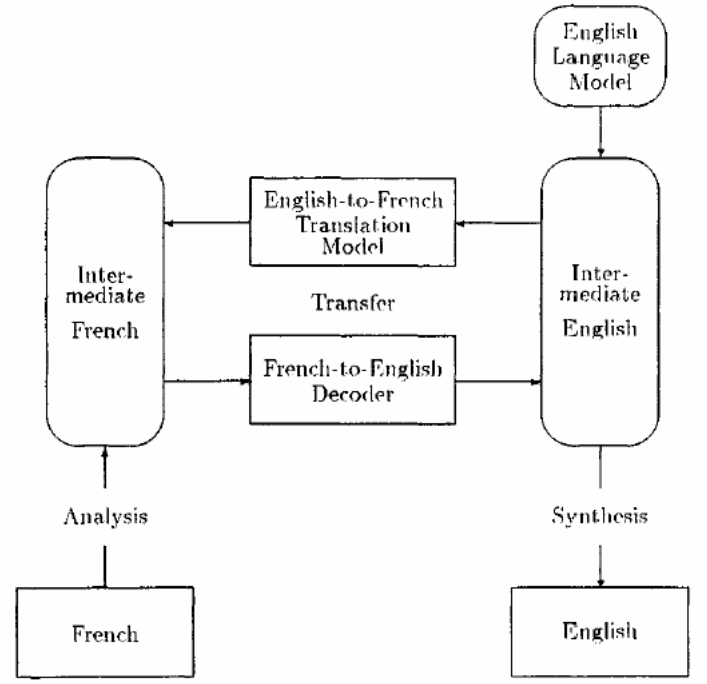

Figure 1.4: The "translation pyramid" adapted to SMT (Brown et al., 1992).

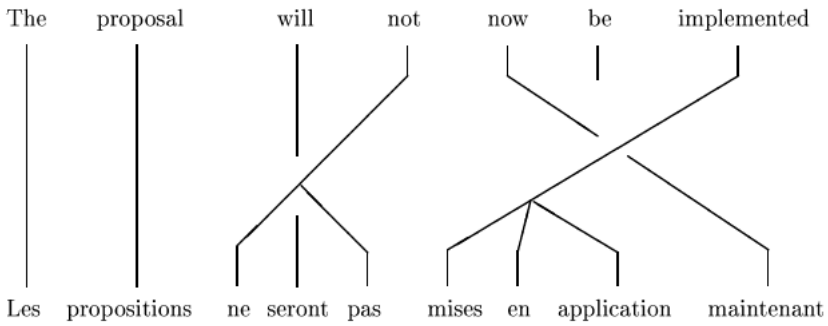

Figure 1.5: Example of a word-to-word alignment (Brown et al., 1990).

The translation model $P(f|e)$ is based upon the following theory about the translation process (Brown, Pietra, Jelinek, Mercer, & Roossin, 1988), (Brown, Cocke, et al., 1988):

1. Partition the source text into a set of fixed locutions.
2. Use a glossary plus contextual information to select the corresponding set of fixed locutions in the target language.
3. Arrange the words of the target fixed locutions into a sequence forming the target sentence.

Thus, the translation model's task is to capture the translation of source locutions into the target language, as well as changes in word-order. Brown et al. (1990) suggest to model translation on the basis of *alignments*, regarding the translation probability $P(f|e)$ as the sum of the probabilities of all possible alignments of the two sentences:

$$P(f|e) = \sum_{a} P(f, a|e)$$

According to Brown et al. (1990), an alignment is a mapping between the source and target sentence, which indicates the *origin* in the target sentence of all the words in the source sentence (see Figure 1.5 for an example). Conceptually, one can separate the *alignment model* from the *lexical model*, which indicates the probability by which one locution is translated into another. Changes in word order between source and target language are captured by the alignment model's *reordering component*.

The complexity of reordering crucially determines the complexity of the *search problem* of SMT (arbitrary word ordering is NP complete). Brown et al. (1993) use Dempster, Laird, and Rubin (1977)'s Expectation Maximisation (EM) algorithm[2] to extract the parameters from a translation corpus. Generally, the search space of (more complex) alignment models is too big to be searched exhaustively.

1.1.1 Word-based SMT

Avoiding step 1, the partitioning into fixed locutions, Brown et al. (1990)'s translation model is based on word alignment. Here, the source sentence is regarded as generated from the target sentence *word by word*. The disadvantage is that at most one target word can be aligned with each source word. The number of source words generated from a target word Brown et al. (1990) model by means of *fertility* probabilities $f(e_i)$. Brown et al. (1990)'s reordering component consists of a *distortion model* $d(i, j|l)$. Thus, the probability of d is dependent on the relative positions of the source and target words (j and i) and the target sentence length l (Brown et al., 1990). Brown et al. (1993) present four different word-based translation models, whose alignment models are of increasing complexity, which are often referred to as "the IBM models".

[2] Although this is not indicated in (Brown et al., 1993).

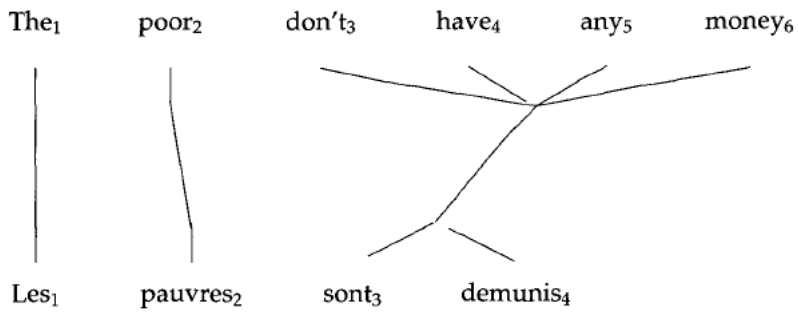

Figure 1.6: Example of a sentence pair that cannot be aligned word-by-word (Brown et al., 1993).

Unfortunately, word alignment is inadequate even for translation between closely related languages such as French and English:

> "We have serious problems in sentences in which the translation of certain source words depends on the translation of other source words. For example, the translation model produces *aller* from *to go* by producing *aller* from *go* and nothing from *to*. Intuitively we feel that *to go* functions as a unit to produce *aller*." (Brown et al., 1990)

Moreover, the exact alignment of the individual words within phrase translations is often impossible to determine even for humans (see Figure 1.6).

Another "prime deficiency" of Brown et al. (1993)'s approach is the reordering component (Fox, 2002). The distortion model is linguistically uninformed as it "pays little attention to context and none at all to the higher-level syntactic structures" (Fox, 2002). For illustration, consider the following analogy:

> "to make a comparison with clothes, to localise what corresponds to the left shoulder of a shirt on, say, a jacket, one does not take material from the left shoulder of the shirt, unweave it, weave it back again in a different way, and then patch it somewhere on the jacket. (Lepage & Denoual, 2005a)"

As a matter of fact, structural differences between languages increase the search-space along all parameters. For example, as shown in Figure 1.7 the language pair Japanese-English has considerable differences in word-order. As a consequence, beam search decoders are often stuck in sub-optimal solutions, when applied on structurally different language pairs, such as Japanese-English (Watanabe & Sumita, 2003). In a large-scale experiment on the language pair French-English, word-based SMT proved quite "able to select the right target words", but in contrast "there is little prospect of getting these in the right order" (Way & Gough, 2005).

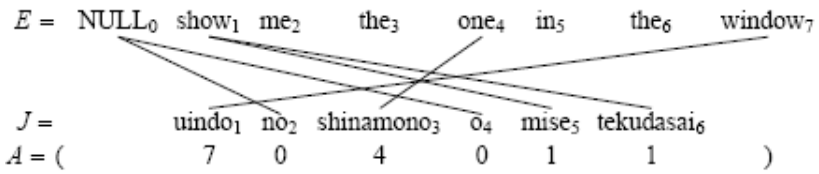

Figure 1.7: Word-by-word alignment for an example sentence pair Japanese-English: Discrepancies in word-order enlarge the search space (Watanabe & Sumita, 2003).

1.1.2 Phrase-based SMT

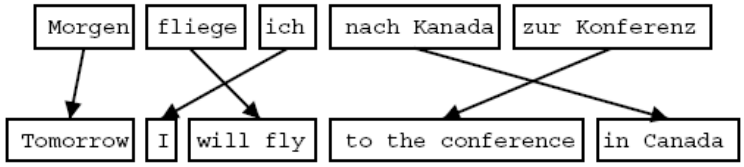

Figure 1.8: Example of phrase-based alignment (Koehn, 2004).

In contrast to word alignment, *phrase alignment* does not align individual words, but "groups of words in the source language that function as a unit in translation"(Brown et al., 1990) (see Figure 1.8). This is motivated by a phenomenon called *phrasal cohesion* that has been observed since the very beginning of statistical machine translation:

> "Words form phrases in the target sentence that are translations of phrases in the source sentence and that the target words in these phrases will tend to stay together even if the phrase itself is moved around" (Brown et al., 1990).

In *phrase-based translation*, the input sentence S is thus segmented into phrases s_i, as basic unit of translation. Notably, these phrases are *sequences of consecutive words* (Koehn, Och, & Marcu, 2003), which means they "can be any substring and not necessarily phrases in any syntactic theory" (Chiang, 2005). Since the lexical model as well as the reordering component work at phrase level, the phrases are translated and reordered as a whole. For example, Koehn (2004) performs reordering of the translated phrases into the target language according to some distortion model d. Further, a cost factor ω larger than 1 biases towards longer translation phrases. Thus, the translation model $P(S|T)$ is decomposed into

$$P(s_1^I|t_1^I) = \Pi_{i=1}^I \phi(s_i|t_i) d(a_i - b_{i-1})$$

and the overall decoder equation (Koehn, 2004):

$$T_{best} = argmax_T P(T|S) = argmax_T P(S|T)P(T)\omega^{length(T)}$$

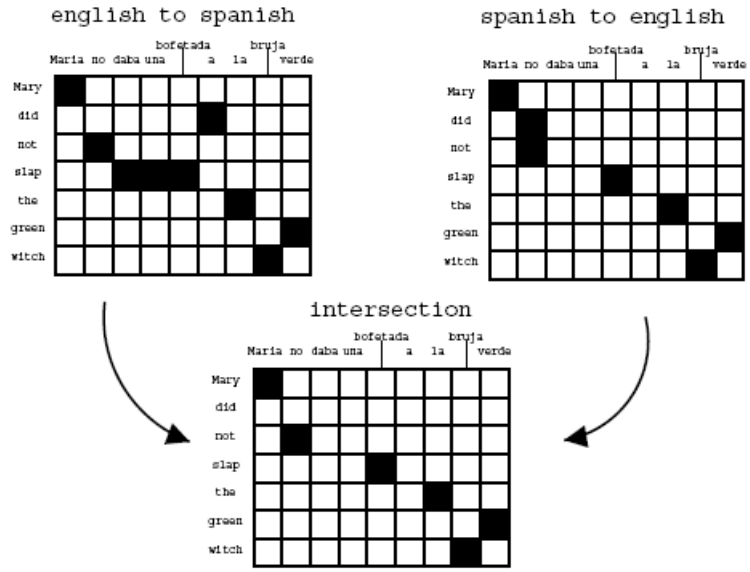

Figure 1.9: Illustration of bidirectional alignments and their intersection (Koehn, 2004).

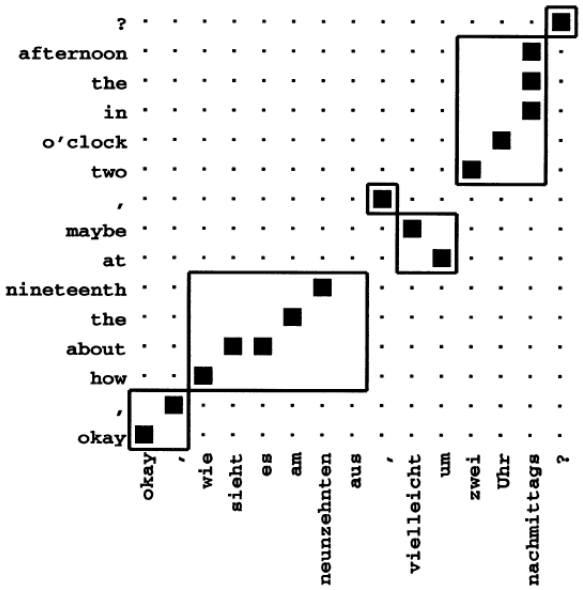

Figure 1.10: Phrase segmentation based on word alignment in the alignment template approach (Och & Ney, 2004).

Phrase translations can be learned directly in Marcu and Wong (2002)'s joint probability model, or extracted from bi-directional word alignments (Och & Ney, 2004). If phrases are extracted, the alignment quality can be improved if word alignment is performed *bi-directionally* (see Figure 1.9): The union of both alignments yields a high-coverage low-precision alignment, while the intersection gives a low-coverage high-precision alignment. As shown in Figure 1.10, Och and Ney (2004) then extract all phrases that are *consistent with the word alignment*, i.e. all words must be aligned exclusively to words *within* the phrase (Och & Ney, 2004). After extraction, the probability of the phrase translations can be estimated by relative frequency (Koehn et al., 2003). Koehn et al. (2003) compare the two methods experimentally, and find that phrase extraction based on word alignment yields a slightly higher BLEU score in translation. Och and Ney (2004) claim that it is also easier to implement than the joint model.

Phrase-based SMT relies on the *cohesion* of contiguous phrases (sequences of words) during translation (Fox, 2002). If a phrase is common enough to have been observed in training data (as *direct match*), it can be robustly translated without needing to be reconstructed (Quirk & Menezes, 2006). Thus, phrases provide a *local context*, which – depending on the phrase length – can contain word reorderings ("yellow house – maison jaune"), fixed expressions ("kicked the bucket – starb"), word deletions/insertions ("to school – in die Schule") as well as lexical disambiguation ("la maison – the house", not "le maison", even though "le–the" might be more frequent). Thus, in contrast to word-based SMT, phrase-based SMT can robustly translate such phenomena, as long as they are sensitive to the local context provided by the phrase. Hence, it is unsurprising that the translation quality is "considerably better" than that obtained by word-based models (Groves & Way, 2005).

While the move from words to phrases as the basic unit is a major advancement, the expressive power of phrase alignment is still limited by its constraint to contiguous phrases:

> "If a consecutive phrase in one language is translated into two or three non-consecutive phrases in the other language, there is no corresponding bilingual phrase pair learned by this approach." (Och & Ney, 2004)

Simard et al. (2005) describe the use of non-contiguous phrases at the example of the following English-French translation pair:

(1.1) "Mary switches her table lamp *off* → Mary éteind sa lampe de chevet"

In French, the verb "switch" can be translated as "allumer" (to switch on) as well as "éteindre" (to switch off). Word-based alignment would have a 50-50 bet to get the verb-translation right, as it aligns "off" to the empty word ϵ. Phrase-based translation could only do better if it happened to have learned the whole phrase containing the discontiguous phrase, i.e.

(1.2) "switches her table lamp off – éteind sa lampe de chevet"

The contribution of discontiguous phrases to translation performance has been shown by Bod (2007). The fact that neither Och and Ney (2004) nor Marcu and Wong (2002) manage to extract non-contiguous phrases does not mean that these models were not capable of extracting discontiguous cases *in principle*. Rather, in practice this causes the alignment model to become too large and noisy. Indeed, according to Simard et al. (2005), non-contiguous phrases cause the number of extractable sentences to grow exponentially with the sentence size, while it only grows quadratically with exclusively contiguous phrases. Unfortunately, the number of extracted phrases has a "direct impact" on the computation time of a translation (Simard et al., 2005). Thus, the main challenge consists in finding those phrases that are most beneficial for translation. Then, the remaining problem consists in reordering these discontiguous phrases.

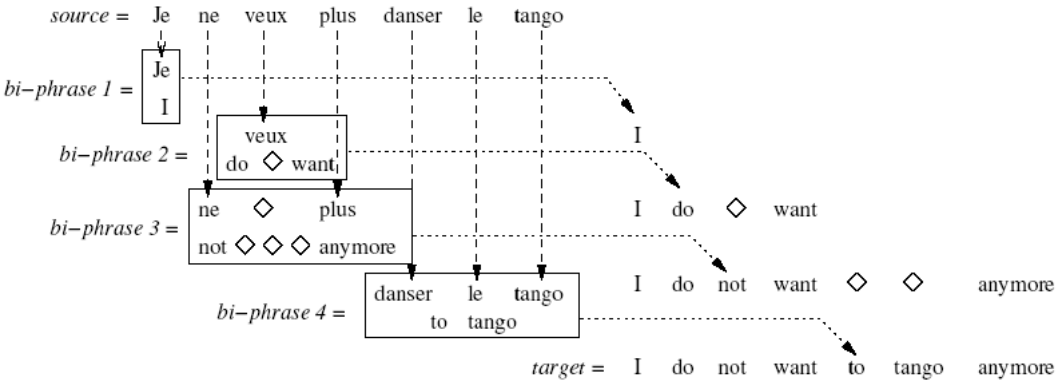

Figure 1.11: The combination of discontiguous phrases in phrase-based SMT (Simard et al., 2005).

As context-sensitivity in phrase-based SMT is restricted to the local context provided by the phrases, there is no convincing strategy for the global reordering of the phrases (Quirk & Menezes, 2006). Koehn et al. (2003)'s distortion models are essentially the same as Brown et al. (1990)'s, in that reordering exclusively depends on the relative positions of the source and target phrases in the sentence. Thus, the only potential advantage would stem from the fact that changes in word-order are modelled above the phrase level, rather than the word level. However, according to Quirk and Menezes (2006) the resulting reduction of the search space is not significant, as exhaustive search is still impractical for most sentences. Further, since phrases are non-constituents, they do not provide a basis for linguistic generalisation to capture structural discrepancies (Quirk & Menezes, 2006). Under these circumstances, it seems questionable, if changes in phrase-order can reflect structural discrepancies between source and target language. Further, the phrases tend to be rather small (e.g. "up to three words" according to Koehn et al. (2003)), and hence cannot be expected to contain much syntactic information within the local context. We conclude that phrase-based models lack a representation of syntactic information almost as much as the poorer word-based models.

1.1.3 Syntax-based SMT

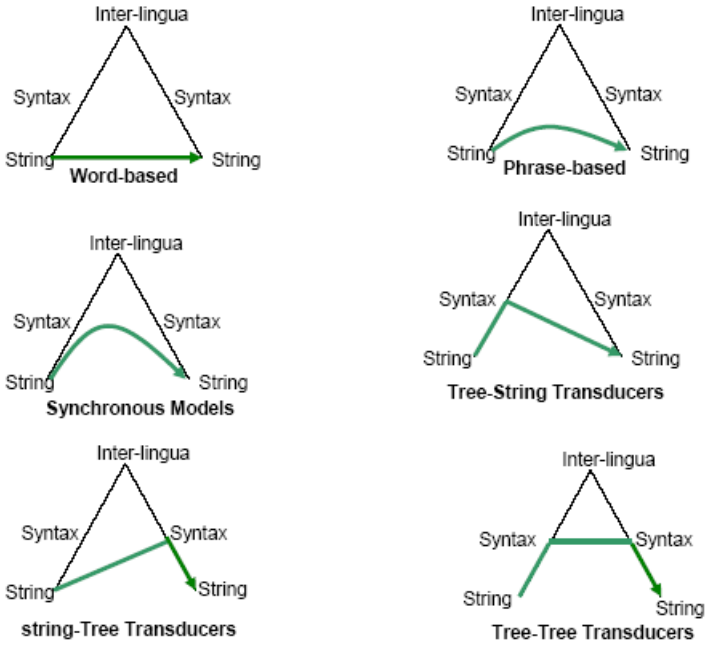

Figure 1.12: Syntax in SMT (Ahmed & Hanneman, 2006).

Syntax-based SMT aims at modelling structural or syntactic aspects of language within a statistical framework.

> "Adding syntax back to the translation process is not an innovation, rather it is just fall back to an old standard practice in human-intensive machine translation systems. However, the main breakthrough here is that no human labor is required beyond those efforts already investigated to build monolingual syntactical parsers." (Ahmed & Hanneman, 2006)

This approach potentially offers two advantages over phrase-based SMT, namely to model a) syntactically motivated constraints on word order and b) long-distance dependencies between non-contiguous phrases (Huang & Knight, 2006).

Within the source-channel SMT model, syntactic information can be incorporated in the target language model as well as the translation model. Firstly, the target language model can be enhanced with target language syntax. For example, the n-gram model can be replaced by a syntactic parser. The probability of a target sentence is then calculated as the sum of the probabilities of all its parse trees. Charniak, Knight, and Yamada (2003) show that this can improve the grammaticality of the target sentences

as well as the translation accuracy. However, this includes syntax merely in a *post-hoc* reranking process, not *at the core* of translation (Way & Gough, 2005).

Secondly, the translation model can be enhanced with target as well as source syntax (see also Figure 1.12). Yamada and Knight (2001) incorporate source language syntax, as they translate an input tree into an output string. Such a *string-to-tree* translation model is computationally similar to a parsing model. As shown in Figure 1.13, Yamada and Knight (2001)'s channel operations are: reordering child nodes, inserting extra words at each node, and translating leaf words on each node of the parse tree. Cowan, Kucerova, and Collins (2006) also incorporate the parse tree of the target language. Such a *tree-to-tree* translation model maps parse trees in the source language to parse trees in the target language. This has two potential advantages: improved output grammaticality by modelling the syntax of the target language and a detailed model of correspondences between the parse trees of the source and the target sentence.

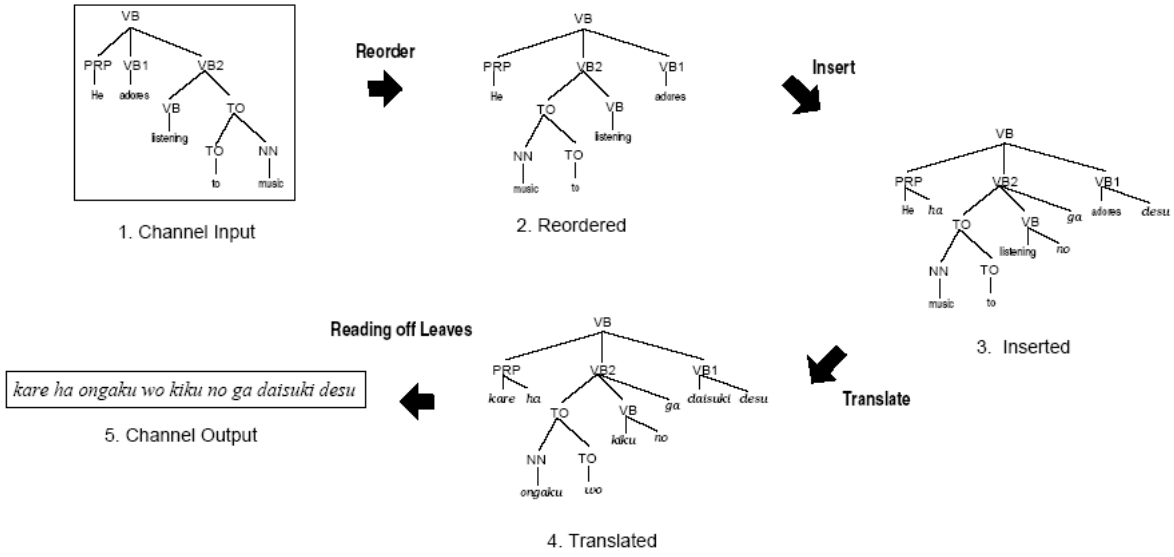

Figure 1.13: Translation of a source tree into a target string (Yamada & Knight, 2001).

Unfortunately, parsing models (having been developed to capture dependencies within a single language) cannot be expected to automatically reveal translation correspondences across languages (Tinsley, Hearne, & Way, 2007). Identical parsing schemes will not produce *parse-tree isomorphism* (bi-directional one-to-one mapping of the nodes) when applied to different languages. This makes it more difficult to e.g. employ syntactically motivated constraints. For example, straightforward experiments limiting phrase-based SMT models to learning syntactically motivated phrases have led to a decrease in performance (Koehn et al., 2003). While this might largely be the result of a severe reduction of overall phrase quantity, the fact that not all potentially useful phrase-pairs are syntactic constituents cannot be ignored: e.g. "there is – es gibt – il y a", "I will – ich

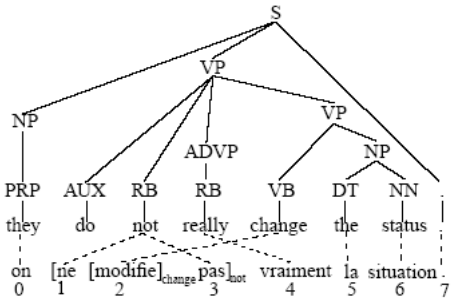

Figure 1.14: Phrase alignment crossing due to negation (Fox, 2002).

werde – je vais", etc. As another example, Yamada and Knight (2001) restrict the ordering of the target language words based on syntactic information of the source language upon *child reordering* (see Figure 1.13). This means that all children of the nodes in the source syntax tree can be reordered before they are translated independently. However, it turned out that child reordering cannot capture all structural discrepancies to be found in manual word-alignments on French-English (Fox, 2002) – and more problems are to be expected with less closely related language pairs. For example, as shown in Figure 1.14, child reordering cannot handle the correct word order for negation. Rather, a language-dependent solution would be required to cover systematic structural discrepancies. Generally, there are two ways to overcome this problem: a) a more sophisticated method to *extract* the information required for translation, and b) *reforming* the syntax-model such that information about correlations across languages is more accessible in the first place.

For example, Cowan et al. (2006) extract *aligned extended projections* for each source clause. More specifically, they align every source clause with a slightly modified representation of the extended projection of its translation in the target sentence. As shown in Figure 1.15, an extended projection is a syntactic structure that contains a selected subset of the words in the original sentence: a single content word and at least one function word that is associated with the content word. Cowan et al. (2006) focus on verb-clauses, whose associated function words include modal verbs and wh-words, and – as in the example in Figure 1.15 – auxiliaries (e.g. has) and complementisers (e.g. that). In contrast, all NPs and PPs occurring in the phrases are generalised over. Aligned extended projections are extracted automatically based upon word alignment of NPs and PPs (extracted from statistical word alignment) and the clause structure provided by the parser. For translation, the source sentence is first broken down into clauses, which are translated individually on the basis of the most probable extended projection and then recombined to form the output sentence.

The reliance on syntactic deep-processing can also be regarded as a disadvantage,

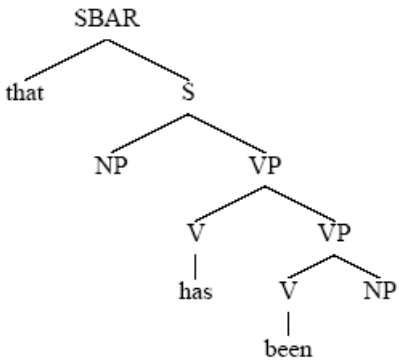

Figure 1.15: The extended production of the clause "that the main obstacle has been the predictable resistance of manufacturers" (Cowan et al., 2006).

as robust syntactic parsers are not always available and parsing errors are an additional source for misperformance. This may be an additional reason why such syntax-based approaches have so far not produced systems that can compete with phrase-based systems in large-scale translation tasks (Chiang, 2007). Recently, new approaches have emerged that no longer rely on linguistic annotation, but only make use of the *formal machinery* of syntax without any linguistic commitment. Strictly speaking, such systems are *formally syntax-based* but not *linguistically syntax-based* because they do not make use of any syntactic theory (Chiang, 2005).

Synchronous context-free grammars (SCFG) are one example of a grammar formalism that specifies relationships between two languages. It is similar to CFG, but generates pairs of related strings instead of single strings. For example, the correspondence between a SVO and and a SOV language can be captured by the following rules:

(1.3) S $\rightarrow$ [(NP VP), (NP VP)]

(1.4) VP $\rightarrow$ [(V NP), (NP V)]

Unfortunately, there is a scalability issue with SCFGs as (in contrast to CFGs) they cannot always be converted to Chomsky normal form, which is why they do not have a practical implementation yet (Ahmed & Hanneman, 2006).

> To learn these synchronous grammars, one needs to model the joint probability of the source and target languages – that is $p(e, f)$ – using hidden variables to account for the missing bitree. An EM algorithm is then used to estimate the required parameters. However this involves a costly E-Step during which the bilingual text is parsed in (n^6) in most formalisms using a variant of the inside-outside algorithm used in monolingual statistical parsers.

Therefore, to scale these systems one needs to impose certain restrictions on the grammar expressiveness to avoid the costly E-Step.

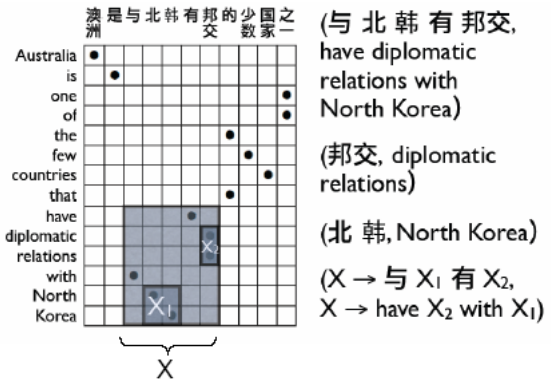

Figure 1.16: Extraction of hierarchical phrases(Chiang, 2007)

Chiang (2005) uses a restricted form of SCFG to extract *hierarchical phrases*, i.e. phrases which contain generalisations of subphrases. These phrases thus add another hierarchical level on top of phrase-based SMT. Being discontiguous, the scopes of hierarchical phrases are larger than that of conventional phrases. Further, they specify the reordering of the sub-phrases within their scope. This makes them more powerful than conventional phrases. As shown in Figure 1.16, hierarchical phrases are induced heuristically from phrase-aligned bitext without linguistic annotation. According to Chiang (2007), hierarchical phrase-based translation is "the first system employing a grammar (to our knowledge) to perform better than phrase-based systems in large-scale evaluation".

1.2 Example-Based Machine Translation (EBMT)

According to Sato and Nagao (1990) "the basic idea of example-based translation is very simple: translate a source sentence by imitating the translation example of a similar sentence in the database". This paradigm has been first proposed by Nagao (1984) who was inspired by the "mechanism of human translation":

> "Man does the translation, first, by properly decomposing an input sentence into certain fragmental phrases (very often, into case frame units), then, by translating these fragmental phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference." (Nagao, 1984).

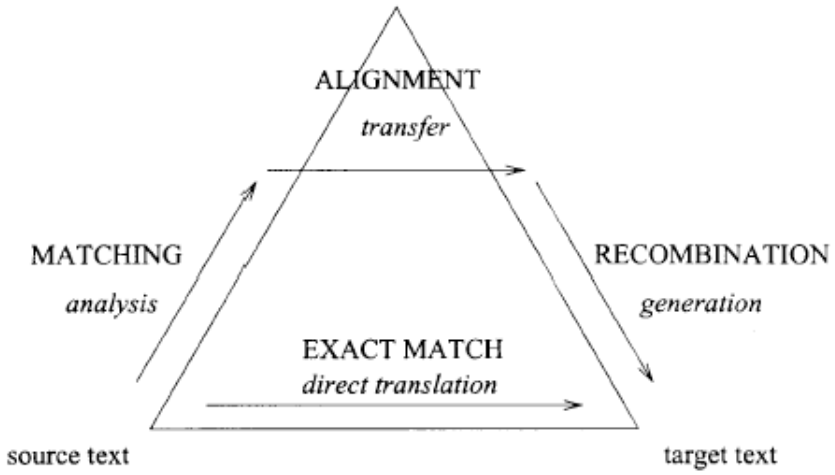

Figure 1.17: Translation pyramid adapted to EBMT (Somers, 1999). While the matching and recombination stages are "very similar", even in terms available of implementation techniques, relating "direct match" to "exact match" is more controversial: "In one sense, the two are alike in that they entail the least analysis; but in another sense, since the exact match represents a perfect representation, requiring no adaptation at all, one could locate it at the top of the pyramid instead" (Somers, 1999).

This statement reflects the three main components of an EBMT system (Somers, 1999):

1. Matching the input sentence against the example base.
2. Transferring fragments of the input sentence into the target language.[3]
3. Re-combining the target language fragments into the target sentence.

Nagao (1984)'s aim was to develop a translation system "based on the fundamental function of language processing in the human brain". Indeed, the concept of EBMT is very similar to human translation, assuming that a translator will also recognise if she has translated a similar sentence before, and then use this as a role model for the new translation (Collins & Cunningham, 1996). However, the original basis for EBMT had been Nagao (1984)'s view on *foreign language learning*:

[3] As in Figure 1.17, the transfer phase is often referred to as "alignment". However, there is an important difference between alignment in EBMT and SMT: EBMT alignment is based on the comparison of two sentences, namely the translation example and input sentence. Accordingly, EBMT does not have to align every word in the senence, but only those words (or phrases) which make up the difference between input and example sentence.

> "A student memorises the elementary English sentences with the corresponding Japanese sentences. The first stage is completely a drill of memorising lots of similar sentences and words in English, and the corresponding Japanese. Here we have no translation theory at all to give to the student. He has to get the translation mechanism through his own instinct. He has to compare several different English sentences with the corresponding Japanese. He has to guess, make inferences about the structure of sentences from a lot of examples." (Nagao, 1984)

Likewise, an EBMT system must be able to recognise similarities and differences between translation examples. Information about the structure of the sentences as well as correspondences between words can then be extracted by the *replacement operation* illustrated in Figure 1.18. If two examples differ by exactly one source and target word, respectively, these new words can be recognised as a new translation pair. Moreover, if such information is already available, the system will be able to translate a new input sentence by modifying the example translation. For example, the system can translate the sentence "A man eats vegetables." on the basis of an example translation of the sentence "He eats potatoes.", given that it already knows how to translate "a man", "he", "vegetables" and "potatoes".

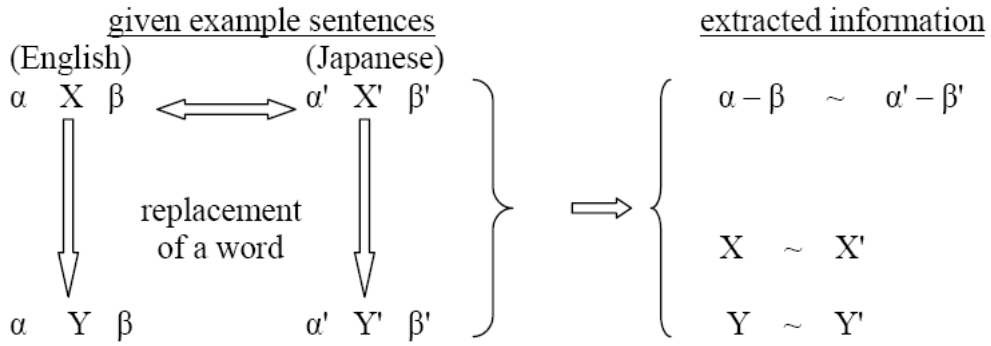

Figure 1.18: Illustration of the replacement operation (Nagao, 1984).

According to Lepage and Denoual (2005b) "the difficulty which is usually faced in translating between some particular languages partly vanishes (at least in theory!)" in consequence of this view. EBMT may thus bridge the *idiosyncratic gap* between structurally different languages (see Figure 1.2), without the need of a global translation model:

> "In contrast to some other approaches to machine translation, namely statistical machine translation, which do not view linguistic data as specific data, we believe that natural language tasks are specific because their data are specific." (Lepage & Denoual, 2005b)

Indeed, EBMT has first been applied on structurally different languages and proposed for particular linguistic phenomena regarded as too difficult for the rule-based approach (Sumita & Iida, 1991).

From the machine-learning perspective, EBMT can be regarded as an instance of *case-based reasoning (CBR)*, which has originally been developed as an alternative for rule-based expert systems:

> "Instead of rules, CBR represents expertise in the form of past 'cases', and new problems are solved by finding the most similar case in the case-base, and using this as a model for the new solution through a process of 'adaptation'." (Somers, 2001)

According to Somers (2001), there is a strong parallel between EBMT and CBR. In particular, both approaches attempt to avoid the knowledge acquisition bottleneck by means of an intuitive problem solving approach based on human problem solving. Further, CBR cases just as EBMT examples can be regarded as "very specific rules" which only apply to distinct situations (Somers, 2001). Accordingly, they share the problem that a case base may contain overlaps in the form of either mutually reinforcing or conflicting examples (Somers, 1999).

Another problem in EBMT is that the replacement operation will not automatically produce correct translations if applied on *any* translation pair and input sentence:

> "The snag is that even though it may be trivial to adapt an example to suit the new problem at hand, this does not guarantee that the corresponding changes can be propagated across to the target language solution." (Collins & Cunningham, 1997)

Even if the replaced words are translated correctly, the sentence may be syntactically or semantically incorrect after recombination. This problem is called *boundary friction* and can be caused by syntactic as well as semantic discrepancies between source and target language. As an example, consider the translation of "He eats a banana." on the basis of the example "He eats a sandwich." into German. If we simply replace the word "sandwich" by "banana", this will result in a violation of German gender agreement, as the change of syntactic gender from neutral to feminine also requires a change of the determiner ("ein Sandwich" but "eine Banane"). Similarly, a sentence like "Elephants eat bananas." would require a change of the verb ("fressen" instead of "essen"), due to the subtle semantic difference of the agent (animal instead of human) which is reflected in German but not in English. Thus, the application of the replacement operation must be restricted to suitable example translations for a given input sentence.

The selection of suitable example translations from the example base is called *retrieval* and is usually performed on the basis of a *similarity metric* comparing input sentence and example source. Nagao (1984) specifies the following two criteria:

1. The input sentence must be *syntactically similar* to the example source.

2. Those words that are replaced during the modification of the example sentence must actually be *replaceable*.

For the latter Nagao (1984) proposes to use a thesaurus, such that words are only replaced by near synonyms:

> "The replaceability of the corresponding words is tested by tracing the thesaurus relations. If the replaceability for every word is sufficiently sure, then the translation sentence of the example sentence is changed by replacing the words to the translation words of the input sentence." (Nagao, 1984)

Recently, a great variety of different EBMT techniques have been developed.

> "Within EBMT there is [...] a plethora of different methods, a multiplicity of techniques, many of which derive from other approaches: methods used in RBMT systems, methods found in SMT, some techniques used with translation memories (TM), etc. (Hutchins, 2005).

Lepage and Denoual (2005a) claim to have built "the purest ever" EBMT-system, because

> "it strictly does not make any use of variables, templates or training, does not have any explicit transfer component, and does not require any preprocessing of the aligned examples" (Lepage & Denoual, 2005a).

However, this is exactly what many other (potentially less "pure") EBMT-systems make use of.

According to Somers (1999) major differences between EBMT systems manifest in the *example base*. In order to avoid overlapping (and in particular conflicting) examples, some approaches filter the example base, or even construct it manually, while others rely on a similarity metric to reduce the number of matches (Somers, 1999). There have also been efforts to combine several retrieved translation examples (Sato & Nagao, 1990). Translation examples are usually stored at sentence level, or a smaller grain size. They may be stored as plain strings (Lepage & Denoual, 2005b), or enhanced with linguistic annotation (Carl, 1999), or even tree structure (Sato & Nagao, 1990). As illustrated in Figure 1.19, this additional information can be used for abstraction of the actual translation examples. However, according to Carl and Hansen (1999), there is a trade-off between coverage and reliability:

> "the more an MT system is able to decompose and generalise the translation sentences, translate parts or single words of it and to recompose it into a target language sentence, the broader is its coverage and the more it loses translation precision" (Carl & Hansen, 1999)

As illustrated in Figure 1.20, a translation system that employs a low level of abstraction and a coarse granularity of the matching fragments will achieve the highest reliability. However, such a system will be in trouble when faced with an input sentence that is different from those sentences in the training data:

> "To attain a certain degree of creativity, an appropriate degree of abstraction, and an appropriate degree of decomposition is required" (Carl, 1998).

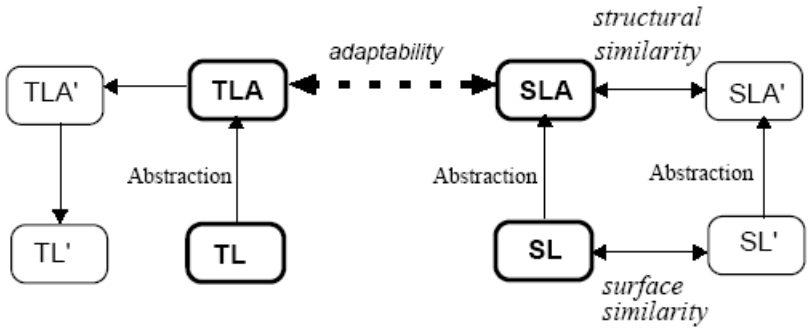

Figure 1.19: Abstraction in EBMT (Collins & Cunningham, 1996). TLA and SLA are the abstracted representations of the translation example TL and SL. SL' is the input sentence to be translated to TL'. A good translation example should be similar to SL', but more importantly its abstraction SLA must be similar to SLA'.

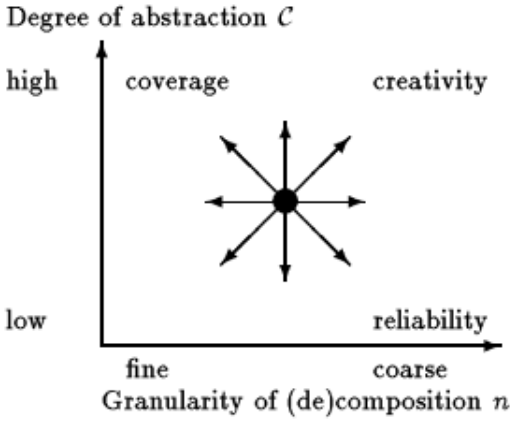

Figure 1.20: Trade-off between coverage and reliability (Carl, 1998): A high degree of abstraction and fine decomposition granularity leads to a high coverage (upper left). A low degree of abstraction and coarse granularity leads to a high reliability (lower right).

The way the examples are stored in the example-base has a great influence on the *matching* phase, as different storing methods allow for different *similarity measures*.

Structural matching of trees is computationally more complex than *string-based* matching against the source side of a template. Moreover, since the template already specifies the translation of the remainder of the sentence and the word order, the translation process mainly consists of translating those phrases that match on the variables of the template (Kaji, Kida, & Morimoto, 1992). In string-based matching, computational resources can be saved if whole words rather than individual characters are taken as the smallest unit (*word-based* matching). Corresponding pieces below the word level can only be detected by the more expensive *character-based* matching. Lepage and Denoual (2005a) show that translation correspondences can be distributed over the entire string of the translated sentence, as in:

(1.5) "Atraversó el río flotando – It floated across the river."

In practice, the best method depends on the language pair. For example, highly inflecting or agglutinative languages particularly benefit from morphologic annotation.

In the following two sections, we contrast the so-called "pure" approach with "two-phase EBMT", which divides the overall process into a preprocessing and a translation phase.

1.2.1 "Pure" EBMT

I'd like to open these windows.	:	*Could you open a window?*	::	*I'd like to cash these traveler's checks.*	:	*Could you cash a traveler's check?*
↕		↕		↕		↕
Est-ce que ces fenêtres, là, je peux les ouvrir?	:	*Est-ce que vous pouvez m'ouvrir une fenêtre?*	::	*Ces chèques de voyage, là, je peux les échanger?*	:	*Vous pouvez m'échanger un chèque de voyage?*

Figure 1.21: Two proportional analogies that correspond in English and French (Lepage & Denoual, 2005a).

Lepage and Denoual (2005a) claim to have built "the purest EBMT system ever":

> "Our system definitely positions itself in the EBMT stream, however it departs from it in one important aspect: it does not make any use of explicit symbolic knowledge such as templates with variables, and it does not produce any template either. Direct use of bicorpus data in their raw form is made, without any preprocessing." (Lepage & Denoual, 2005b).

The translation system relies entirely on a "specific operation" called *proportional analogy* which reflect the (Saussurian) systematicity of language by exhibiting commutations of sentence fragments:

> "The human interpretation of proportional analogies between sentences is that some pieces of the sentences commute with other pieces, so that human beings perceive it as a kind of parallel replacement." (Lepage & Denoual, 2005b)

For example: "I'd like to open a window." is to the sentence "Could you open a window?", as the sentence "I'd like to cash these traveller's checks." is to the sentence "Could you cash a traveller check?". According to Lepage and Denoual (2005a)

> "any sentence in any language may be cast into a wide number of such proportional analogies that form a kind of meshwork around it" (Lepage & Denoual, 2005b).

However, for the purpose of machine translation, only those analogies which are *corresponding* in source and target language can be used. Figure 1.21 shows an example of such a corresponding analogy in English and French. In order to translate a new input sentence, an *analogical equation* is formed, the resolution of which yields the translation of the input sentence. For example, in order to translate the sentence "Could you cash these traveller checks?", the following analogical equation could be formed:

English: I'd like to open these windows. : Could you open a window? :: I'd like to cash these traveller's checks. : x

$\Rightarrow$ x = Could you cash a traveller's check?

French: Est-ce que ces fenêtres, là, je peux les ouvrir ? : Est-ce que vous pouvez m'ouvrir une fenêtre ? :: Ces chèques de voyage, là, je peux les échanger ? : x

$\Rightarrow$ x = Vous pouvez m'échanger un chèque de voyage ?

Figure 1.22 sketches a Prolog implementation of this translation algorithm. An important feature of the analogy solving algorithm (Lepage, 1998) is *character-based* rather than word-based processing:

> "Approaches that adopt the word as the unit of processing neglect the fact that corresponding pieces of information in different languages are indeed distributed over the entire strings and do not necessarily correspond to complete words." (Lepage & Denoual, 2005b)

Unfortunately, this also brings about a high complexity of the translation algorithm. However, since the complexity is "basically quadratic in the size of the examples" it can be reduced by means of "on-the-fly" selection of the example pairs (Lepage & Denoual, 2005b).

```
% base de connaissance pour la traduction
traduction(s1,ŝ1) .
traduction(s2,ŝ2) .
        ⋮
traduction(sn,ŝn) .
% prédicat de traduction
traduction(D,D̂) :-
    traduction(A,Â),
    traduction(B,B̂),
    analogie(A,B,C,D),
    traduction(C,Ĉ),
    analogie(Â,B̂,Ĉ,D̂),
    assert(traduction(D,D̂)).
```

Figure 1.22: Prolog implementation of the algorithm for translation (Denoual, 2006).

The overall performance of the system is comparable to that of word-based SMT (Denoual, 2006)[4]. However, the system's *low coverage* (less than 0.5 in the experiments in (Lepage, 2005)) poses a serious problem. According to Lepage and Denoual (2005b), this cannot be attributed to their underlying assumptions that analogies of form correspond to analogies of meaning, which seems basically confirmed[5]. Rather,

> "the chance that a sentence will be translated depends crucially on the relevance of the examples covering the domain of the sentences to be translated" (Lepage, 2005).

[4] "Ses performances sont comparables en termes de mesures BLEU et NIST, à celles d'un système statistique utilisant un modèle IBM4 fonctionnant sur les mots (sans les *phrases*)" (Denoual, 2006).

[5] As experiments showed, the proportion of analogies of forms which are *not* analogies of meaning (less than 4%) seems "too small to seriously endanger the quality of the results obtained during translation." (Lepage & Denoual, 2005b) (see Figure 1.23 for an example)

For example, in the experiments reported in (Lepage & Denoual, 2005b), the system formed on average "between half a million and one million analogical equations (687, 641)" to translate one sentence, but it can solve only 28% of these successfully.

Nevertheless, "pure EBMT" constitutes an interesting alternative to SMT, because it implements the principle of *lazy processing* (Lepage & Denoual, 2005b). While SMT performs eager offline learning during the preprocessing phase, this approach does not have any preprocessing phase:

> "There is simply no preprocessing phase because the bicorpus is merely loaded into memory as is before its use in translation, so the preprocessing phase is at a cost of zero!" (Lepage, 2005)

Further, the system even increases its performance during translation (online learning), as it adds newly discovered translation pairs to the database (see Figure 1.22). In particular, all alignment information (below the sentence level) is left *implicit* until needed:

> "the choice of a correct translation is really left to an implicit use of the structure of the target language, and does not imply any explicit transfer processing" (Lepage & Denoual, 2005b)

This approach offers a great flexibilxity in dealing with the specific characteristics of previously unseen input data and at the same time may avoid a great amount of unneeded processing. While in SMT the burden of computation is unequally balanced towards the preprocessing phase rather than the translation phase, this approach balances it (even more unequally) the other way (Lepage, 2005). However, in practice an expensive preprocessing phase (which must only be performed *once*) is probably the lesser of the two evils if the alternative leads to long translation times.

Could you tell me how to fill this from.	:	*Could you tell me how to fill this form.*	$\not:$	*Where is the conference centre?*	:	*Where is the conference center?*

Figure 1.23: An analogy of form that is *not* an analogy of meaning (Lepage & Denoual, 2005b).

1.2.2 Two-Phase EBMT

Two-phase EBMT divides the overall process into a *learning* and a *translation phase* (see Figure 1.24). The learning phase consists of *preprocessing* of the raw translation examples, or – in other words – extracting a *translation grammar* from the raw example base. During the translation phase, an abstract example from the translation grammar is re-instanted with elements from the input sentence. According to Carl (2001), recent developments in EBMT show a trend towards two-phase EBMT:

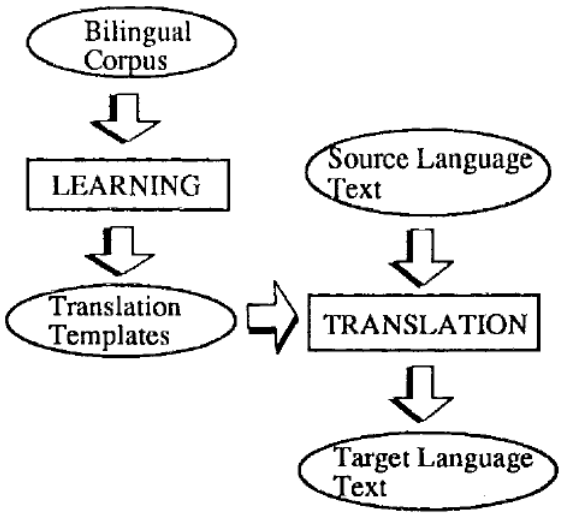

Figure 1.24: Two-Phases EBMT (Kaji et al., 1992).

> The advantage of translation grammars is 1) that they can be induced offline thus reducing computation time in the working-phase of the system and 2) that consistency of the alignments can be checked and handled more easily. (Carl, 2001)

The translation grammar often consists of *translation templates*. Such a template is obtained by generalisation over some of the words in the source and target sentence. These words will then be replaced by variables, which will typically be aligned. In addition, the variables can also be *typed* so as to retain (part of) the linguistic information of the replaced string. For example, the sentence pair

(1.6) John loves Mary → John liebt Mary

could be generized as follows:

(1.7) (...) loves (...). → (...) liebt (...)

After alignment, the template would be:

(1.8) X loves Y → X liebt Y

A typed template couldm retain POS-information:

(1.9) $X_{<NP>}$ loves $Y_{<NP>}$ → $X_{<NP>}$ liebt $Y_{<NP>}$

Cicekli (2005) have shown that type constraints reduce the amount of incorrect translations produced by the application of templates in unrelated contexts. Typed translation templates enable a *unified treatment* of various kinds of translation knowledge (Kaji et al., 1992)

The individual rules of a translation grammar resemble translation rules in RBMT, except that they are extracted automatically and thus tend to have fewer constraints (McTait, 2001). The degree of resemblance is higher in approaches that heavily rely on linguistic resources and lower in so-called *language-neutral* approaches. Templates have been induced from translation examples annotated with parse trees (Kaji et al., 1992) or linguistic bracketing (Carl, 2001). In language neutral approaches, templates have been computed from the comparison of pairs of translation samples (Cicekli & Güvenir, 2001), or induced from individual examples based on word co-occurrence information (McTait, 2001). In the remainder of this section, we take a closer look at these language-neutral approaches.

Cicekli and Güvenir (2001) propose two template learning algorithms, one generalising over the similarities and the other over the differences of two translation examples. *Similarity translation template learning* (STTL) is essentially an application of Nagao (1984)'s replacement operation, as similar parts of two source sentences are assumed to correspond to each other. *Differences translation template learning* (DTTL) can be regarded as its complement, as here different parts of two source sentences are assumed to correspond, given that the remainder parts are similar (or can be aligned through additional information of lexical translation). Cicekli and Güvenir (2001) use a morphological analysis, in particular to be able to deal with the agglutinative source language Turkish. For example, the following pair of examples contains exactly one similarity (underlined):

(1.10) I break+PAST the window → pencere+ACC kir+PAST+1SG

(1.11) You break+PAST the door → kapi+ACC kir+PAST+2SG

From this, DTTL can thus learn the following templates:

(1.12) break+PAST the → +ACC kir+PAST

(1.13) I X_1 window → pencere X_1 +1SG

(1.14) You X_1 door → kapi X_1 +2SG

Additional templates can be learned by STTL, provided that one of the two differences in the string are already known to be translations of each other. For example, if it is known that "I" corresponds to "1SG" and "you" corresponds to "2SG", the following templates can be learned:

(1.15) window → pencere

(1.16) door → kapi

(1.17) X_1 break+PAST the X_2 → X_2 +ACC kir+PAST X_1

As shown in this example, translation templates can contain lexical translations as well as grammatical correspondences.

Rather than using the comparison of two translation examples, McTait (2001) induces translation templates from *individual* examples making use of word co-occurrence information. McTait (2001)'s focus is on generating translation templates capturing discontiguous phrases such as shown in Figure 1.25:

> "Few, if any, of the existing approaches cater for the fact that translation phenomena are not always bijective and that translation relations of a nature other than 1:1 exist" (McTait, 2001)

In order to achieve this, template generation involves two phases even before the alignment of variables and words: a monolingual and a bilingual phase. During the *monolingual phase*, words that are occurring in the same sentence (at least twice) are combined to *collocations*. More specifically, McTait (2001) builds up trees of collocations with increasing lengths and decreasing frequencies. During the *bilingual phase*, two collocations are regarded as translations of each other, if they have occurred in exactly the same sentence pairs. The longest collocation is selected from target and source tree that fulfils this criterion. Similar as in (Cicekli & Güvenir, 2001), templates are generated from the extracted information in two ways: The first kind of template is generated by replacing all words belonging to the extracted collocation with a variable. For example, the template in Figure 1.25 could be generated from the following two translation examples:

(1.18) The commission gave the plan up. → La commission abandonna le plan.

(1.19) Our government gave all laws up. → Notre gouvernement abandonna toutes les lois.

The second kind of template is the *complement* of the first, as here all *other* words are replaced. In the example, the resulting rules are thus

(1.20) The commission (...) the plan (...). → La commission (...) le plan.

(1.21) Our government (...) all laws (...). → Notre gouvernement (...) toutes les lois.

Both methods extract two kinds of templates that seem useful for translation: The first kind of template represents a phrase translation, in particular templates 1.15 and 1.16. The second kind of template contains discontiguous phrases (see Figure 1.25), or structural information as in templates 1.12 and 1.17. Given structural information for translation is not easy to obtain, the second kind of template seems most valuable

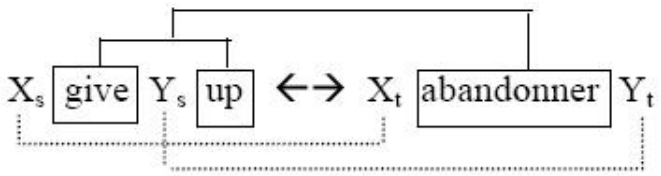

Figure 1.25: A translation template representing a discontiguous phrase (McTait, 2001)

for translation. Finally, both methods also produce templates, such as 1.13, 1.14, 1.20, 1.21, which combine the translation information of seemingly unrelated phrases. Unfortunately, such unrelated phrases may be unlikely to cooccur in a new input sentence. Without further processing, the actual benefit for translation of this third kind of template seems questionable. The problem is, that they are not *representative* of the original example's sentence structure.

1.3 Comparing SMT and EBMT

In this section, we make an attempt to compare the two approaches to MT. Before discussing strengths and weaknesses in Section 1.3.2, we first need to discuss their distinction in Section 1.3.1.

1.3.1 Distinction

The distinction between EBMT and SMT has become a controversial issue recently:

> "Initially, differences between SMT and EBMT were distinct: SMT input was decomposed into individual SL words and TL words were extracted by frequency data (in the 'translation model'), while in EBMT input was decomposed into SL fragments and TL examples (in the form of corresponding fragments) were extracted from the database. More recent developments of 'phrase-based' and 'syntax-based' SMT models have blurred these distinctions." (Hutchins, 2005)

More specifically, EBMT systems have started to integrate statistical operators, and SMT systems have made use of complex representations (Carl, 2006). As a consequence,

> "it is a fact that it is harder than it has ever been to describe the differences between the two methods" (Way & Gough, 2005).

According to Hutchins (2005), it is hard to draw a line

1. between syntax-based SMT approaches and EBMT approaches using syntactic structures to represent the translation examples (Hutchins, 2005).

2. between phrase-based SMT and EBMT in general (Hutchins, 2005).

As Carl (2006) points out, it is even harder to make a systematic distinction between the use of templates in EBMT (Kaji et al., 1992) and hierarchical phrases in SMT (Chiang, 2005). Nevertheless, many attempts have been made to make this distinction.

To begin with, Somers (1999) observes that statistical systems do not store the examples directly, but in terms of precomputed statistical parameters extracted from the overall corpus. The examples themselves are only stored "inasmuch as they occur in the corpus on which the system is based" (Somers, 1999). This has severe consequences for the matching and recombination phase, which in SMT "are implemented in quite a different way" compared to typical EBMT systems (Somers, 1999). Thus, there is a difference concerning the kind of information extracted from the corpus, "and how this is brought to bear in dealing with the new input" (Way & Gough, 2005).

According to Turcato and Popowich (2001), an EBMT system is a translation system that treats the bilingual corpus as a genuine *repository of knowledge* that "could not be stored in other forms". This means that an EBMT-system *must* extract knowledge from the corpus that is only available at runtime. In contrast, SMT systems access the corpus during the preprocessing phase. However, according to Turcato and Popowich (2001)'s criterion, only Nagao (1984)'s translation by analogy "stands out as truly example based". According to Carl (2006), this does not reflect the actual situation:

> we have run-time and compilation-time SMT systems, and we have run-time and compilation-time EBMT systems, which lets us conclude that the distinction run-time versus compilation-time is probably an important detail of the system architecture, but one which does not distinguish major processing paradigm (Carl, 2006)

Hutchins (2005) sees the main difference between SMT and EBMT in their implementation of the *core process* of a MT-system which converts the elements of the input sentence into "equivalent elements" of the target language:

> "MT systems are EBMT systems if the core 'transfer' (or SL-TL conversion) process involves the matching of SL fragments (sentences, phrases, strings) from an input text, the matching of such fragments against a database of bilingual example texts (in the form of strings, templates, tree representations), and the extraction of equivalent TL fragments (as partial potential translations)." (Hutchins, 2005)

However, Carl (2006) also refutes this definition:

> we cannot draw a meaningful distinction between methods being used in EBMT and methods being used in SMT

As an example, he compares (Kaji et al., 1992) and (Chiang, 2005):

> "While the representations are virtually identical, Kaji et al. (1992) refer to their approach as 'example-based' and Chiang (2005) calls his 'statistical'."' (Carl, 2006)

Carl (2006)'s conclusion sheds new light on the EBMT-SMT debate:

> "it is not so much the technical details which distinguishes SMT from EBMT and one system from another. [...] it is mainly due to divergences of the common values which has fragmented the scientific community and which has also been at the core of the putative differences in SMT and EBMT."

In particular, EBMT and SMT researchers have traditionally pursued different *objectives*:

> "while designers of SMT systems seek to optimise an average statistical translation quality, measured on a set of random sentences, followers of EBMT seem to be more interested in increasing the accuracy of the generalisation and reproduction capacities of the example systems" (Carl, 2006)

Further, Carl (2006) regards the issue from the point of view of system theory by Luhmann. In this version of system theory, systems may reproduce itself at "points of re-entry", thereby making use of connectivity operators. Regarding translation examples as linguistic systems, he concludes:

> "EBMT has focussed on the properties of structures suited for translation and the design of their re-entry points, and SMT develops connectivity operators which select the most likely continuations of structures." (Carl, 2006)

In other words, EBMT aims to be *structure preserving*, as it stores the examples' sentence structures implicitly in the examples without modifying them more than necessary. In contrast, for SMT there is a need to decompose the sentence into pieces at least so small that their occurrence is frequent enough for observation in the training data. This leads to a destruction of the sentence structure, which then has to be reconstructed by the statistical model.

In our view, Carl (2006)'s distinction still does not provide a clear classification of real translation systems. The reason for this is that, in practice, EBMT systems also need to work with fragments that have a chance to re-occur in the corpus. And SMT systems have started to increase their fragment size from words to phrases to hierarchical phrases – the latter being hardly distinguishable from translation templates. In sum, SMT and EBMT can be regarded as two lines of research traditions, which already have begun to cross and merge. Strictly speaking, many of the recent translation systems are already a synthesis of the two approaches, although this is rarely acknowledged. The most prominent example for this is (Chiang, 2005), which can also be termed "example-based" if the hierarchical phrases are regarded as (pre-processed) translation examples.

1.3.2 Performance

The actual performance of EBMT and SMT systems has rarely been measured in direct comparison. In fact, Way and Gough (2005)'s and Groves and Way (2005)'s large-scale evaluations on the language pair English-French might be the only ones that have ever been published. Both compare the performance of an EBMT-system to an SMT system given a large amount of training data is available. Way and Gough (2005) use a word-based SMT system, and Groves and Way (2005) a phrase-based SMT system. As to the word-based system, it turned out that the relative performance of the two methods was crucially dependent on the translation direction: While for French to English translation, the SMT system clearly outperformed the EBMT-system, the outcome was reversed for English to French (Way & Gough, 2005). As to the phrase-based system, the best performance was achieved by a hybrid system seeded with all data induced by statistics and the EBMT system (Groves & Way, 2005). Admittedly, these experiments are a comparison of two specific translation systems rather than SMT and EBMT in general. However, they make clear that at the current state of the art, neither approach can be regarded as principally superior to the other.

However, both approaches have their strengths and weaknesses. Traditionally, the following properties have been attributed: Firstly, the minimum amount of training data required for building an SMT system is larger than for EBMT systems (Way & Gough, 2005). Secondly, EBMT is better at dealing with close matches and SMT with remote matches:

> "If a sentence to be translated or a very similar one can be found in the TMEM[6], an EBMT system has a good chance of producing a good translation. However, if the sentence to be translated has no close matches in the TMEM, then an EBMT system is less likely to succeed. In contrast, an SMT system may be able to produce perfect translations even when the sentence given as input does not resemble any sentence from the training corpus. However, such a system may be unable to generate translations that use idioms and phrases that reflect long-distance dependencies and contexts, which are usually not captured by current translation models." (Marcu, 2001)

In our view, the major strenght of the EBMT approach is its distributed information management: Information about *structural discrepancies* are stored *implicitly* in the translation examples. Accordingly, EBMT can build syntax *at its core* (Way & Gough, 2005) without the need of an explicit syntactic translation model. However, pure EBMT has long translation times.

The strength of precompiled EBMT is that during the preprocessing phase extra operations can take place without leading to longer translation times. Unfortunately, preprocessing always entails the risk of losing valuable information. Since the sentence

[6] translation memory = example base

to be translated is an unknown factor during the preprocessing stage, preprocessing can at best make a *probabilistic* distinction between important and unimportant parts of an example translation. A synthesis of EBMT and SMT should thus maintain the distributed information representation, and incorporate statistical preprocessing methods. We will take this idea further in Chapter 3. First, we present the implementation of our EBMT-system in Chapter 2.

2 An EBMT System

In this thesis, we do *not* aim at building a state-of-the-art performing system. Since we need to set up a system *from scratch*, such an objective would be highly unrealistic. Moreover, the performance of any corpus-based translation system is crucially dependent on the amount of bilingual data available, and our resources are limited. Rather, we aim to build a framework for experiments with different EBMT translation methods. This allows to examine specific aspects of the translation process, and compare the performance and the effect of different methodical variations. Accordingly, our system comprises a runtime as well as compiled EBMT approach. We pursue a language-independent approach where a bilingual lexicon is the most important translation resource. If the translation system returns several translations, we take them all. We assume that the best translation among them can be selected by a monolingual language model. We present the implementation of the translation system in Section 2.1 and a basic evaluation of the translation performance in Section 2.2.

2.1 Implementation

Since we pursue a language independent approach, our system will be based on shallow processing. Deep linguistic processing is avoided, as robust parsers are only available for a limited number of languages. In contrast, bilingual dictionaries are available for almost any language pair. Our system thus employs the following translation resources:

1. a sentence-aligned bilingual corpus
2. a lexicon containing phrase and word translations.

We implement two competing methods of EBMT: a runtime translation appraoch as well as a compiled translation approach. Figure 2.1 illustrates the overall system.

Both translation methods follow the classical EBMT procedure (see also Chapter 1): The translation of the input sentence is derived from an *analogous* translation example. The overall translation process consists of the following phases: During *matching* the most similar translation example(s) are retrieved, then each of these *matched sentences* is *analysed* before the translation can be generated by means of *recombination*. Our method generates an individual set of translations from each of the matched sentences.

We will compare a *runtime* approach which relies on unprocessed translation examples, and a *compiled* approach which employs a translation grammar. As described in

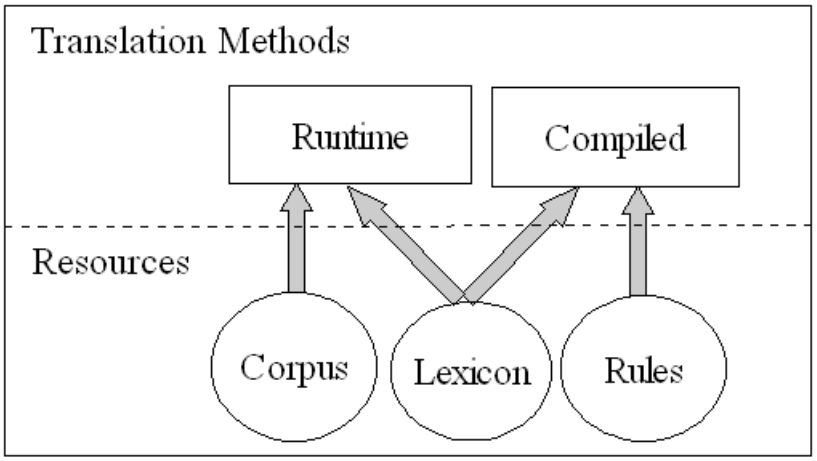

Figure 2.1: The architecture of our EBMT system. The runtime translation method is an instance of "pure" EBMT in the sense that it uses raw translation examples (corpus) and a lexicon. The compiled translation method is an instance of two-phase EBMT because it uses precompiled translation rules instead of the raw corpus. However, both approaches use the corpus to disambiguate the lexicon lookup of phrases contained simultaneously in the input sentence and the example source (see Section 2.1.1).

Chapter 1, the compiled approach makes use of a previously extracted translation grammar consisting of translation templates. In contrast, the runtime approach is exclusively based on the comparison between input and matched sentence, generating an (implicit) template "on the fly".

We describe the runtime approach in Section 2.1.2 and the compiled approach in Section 2.1.3. Both approaches heavily rely on the lexicon, which we describe in Section 2.1.1.

2.1.1 Lexicon

We aimed at building a lexicon that contains translations of words and contiguous phrases. Lexica are an important translation resource, and their great advantage is that they can be obtained easily for virtually *any* language pair. Since bilingual dictionaries are a wide-spread linguistic resource, one can expect that at least one digital dictionary is available for almost any language pair. However, linguistic resources are not the only source for a lexicon, as word and phrase translations can also be automatically extracted from bilingual corpora. Both lexicon sources have advantages and disadvantages: While hand-collected dictionaries are more reliable, they only contain one standard form of each word. Thus, they can only be fully exploited in combination with (error-prone) morphologic processing. Extracted word and phrase translations may not be as reliable,

but they contain different word forms. Moreover, they are automatically tuned to the text type of the text from which they have been extracted.

In our implementation, we combine both lexicon sources. More specifically, we combine freely available online *dictionaries* with a statistically extracted *phrase table*. In addition, we use and even give preference to the translation contained in the retrieved translation example. With so many sources combined, a lexicon lookup typically yields a set of translations rather than a single translation. If a phrase occurs simultaneously in the input sentence as well as in the source sentence of the translation example, we make use of this fact to disambiguate the translation. More specifically, if at least one of the translations we obtain from the lexicon lookup occurs in the example's target sentence, we filter out all translations that do not occur in the example. Finally, if no other translation is available, we take the original word, as this is usually the right translation for a proper noun.

We describe the dictionaries in Section 2.1.1 and the phrase table in Section 2.1.1.

The Dictionary

After a brief search of the web, we decided to use two online dictionaries: Woxikon (http://www.woxikon.com/) and Ectaco (http://www.ectaco.co.uk/). These two dictionaries are freely available, provide the language pair English-Dutch, and can be read out automatically with relatively little effort. They are general-purpose dictionaries (i.e. not for any specific domain).

Together, these dictionaries provide a large number of translations for single words, but hardly contain any phrase translations. We have tried to enhance the dictionaries with phrase lookup by the following algorithm:

1. Look up each word in the phrase individually.
2. Return concatenations of the all retrieved translation alternatives.

However, this (naive) implementation could not be run, because the resulting number of combinations was too large to be handled by our computational resources.

As any freely available online dictionary, our dictionaries have been constructed for human rather than computer use. Accordingly, they have the following shortcomings:

1. The dictionaries are generally unable to deal with any form of inflexion (in particular verb forms or plural forms of nouns).
2. The dictionaries often contain either none at all, or many different translations for function words.
3. General-purpose dictionaries are not tuned to our corpus' domain.

In order to make up for the shortcomings, we combine the dictionary with a phrase table extracted from the corpus.

The Phrase Table

Being extracted from the corpus, the phrase table is perfectly tuned to our domain and contains many (and sometimes exclusively) inflected word forms. We first describe how the phrase table is compiled from the corpus, and then how it is used for translation.

Compilation The small size of our corpus turns the task of automatically extracting a reliable knowledge source from a corpus into a challenge. We collected the phrase table as follows: First, we performed word alignment using GIZA++ (Och & Ney, 2000) in both translation directions. We then took the intersection of the bi-directional alignment in order to achieve the best precision possible. As the resulting alignment was still unsatisfactory, we then extracted only non-crossing 1-to-1 aligned word-sequences. This conservative alignment extraction method seems suited for our small corpus, as it yields phrase translations of a reasonable quality.

Lookup Since the extracted phrases are all 1-to-1 aligned, they are not only phrase translations but at the same time also contain word translations. However, we found that the word translations are not precise enough. Instead, we implemented three different lookup mechanisms for our phrase table (in order of their computational complexity):

Exact match lookup: The phrase table's least expensive lookup method is that of the phrase taken as a whole, but it will only succeed if the phrase table contains a source phrase that is an exact match of the input phrase.

Sub-phrases lookup: Since all phrases in the phrase table are 1-one-1 aligned, we can also lookup phrases (or even words) that are subphrases of the original phrase table entry.

Divided lookup: This method consists of dividing the phrase into n subphrases and looking up each of them individually, thereby taking all possible boundary positions into account. Due to its costs, we only apply this lookup method to variable sequences, i.e. if the number of subphrases n is known in advance.

2.1.2 Runtime EBMT

In the runtime approach, the central element is the comparison between example and input sentence. In this section, we first introduce all phases of the basic runtime EBMT algorithm individually, before going through an example translation. A sequence diagram is to be found in Figure 2.2. Then, we describe two extensions of the basic algorithm, namely cutting the matched sentence and phrase enlargement.

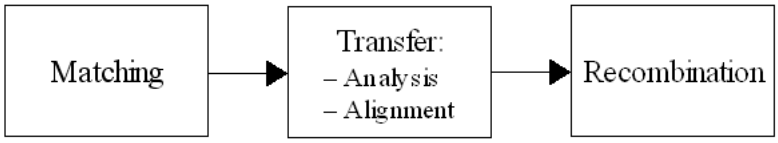

Figure 2.2: Sequence diagram for runtime approach.

Matching As *similarity metric*, we count the number of words the example sentence has in common with the input sentence. The set of matched sentences consists of those example translations with the maximum number of words in common. The algorithm is presented in Figure 2.3.

Transfer During the analysis phase, an implicit translation rule is generated "on the fly". This consists of two parts:

Analysis of the input sentence: By comparing it to the source side of the matched example translation, the input sentence is segregated into identical phrases $i_1 ... i_n$ and analogous phrases $d_1 ... d_m$. *Identical phrases* are those phrases of the input sentence, which are also contained in the matched source (in the right order). *Analogous phrases* are the remainder parts, which are not contained in the example sentence. The analysis algorithm is depicted in Figure 2.4. Note that the algorithm is deterministic and will only return one sentence analysis, even if several ways of matching might be possible.[1]

Alignment: For each d_i in the input sentence we identify the analogous phrase δ_i in the matched sentence source. As shown in Figure 2.5, analogous phrases are identified by *positional correspondence*, i.e. δ_i is situated between the same i-phrases as d_i. The alignment of source and target side of the translation example is established by means of the lexicon. For our translation algorithm, we only need to align the δ-phrases. The translations obtained by lexicon lookup are filtered by the criterion of occurrence in the example's target. If the alignment is unanimous, only one phrase $\hat{\delta'}_i$ will remain.

[1]For example, matching the input sentence "He hands the book to the child" on a rule source "X_1 the X_2" should actually yield two different ways of matching, namely

1. X_1 ="He hands", X_2 ="to the child ."
2. X_1 ="He hands the book to", X_2 ="child ."

However, in the current implementation only the first match is considered, as we do not expect these kind of matches to be very promising in the first place.

Recombination All phrases d_i are translated by lookup in the lexicon which will give a set of translation alternatives D'_i for each d_i. The set of translations of the input sentence is generated by performing a sequence of substitutions on the example target: Every $\hat{\delta'}_i$ is replaced by all possible translations $d'_i \in D'_i$ of d_i.[2] The recombination algorithm is depicted in Figure 2.6.

Here is a (constructed) example that illustrates the translation process described: Suppose our input sentence is

(2.1) "The children only eat sweets but no carrots."

and our corpus contains the translation example

(2.2) "Vegetarians only eat vegetables but no meat. → Vegetarier essen nur Gemüse aber keine Fleischprodukte."

The input sentence would be divided into the identical phrases i_1 ="only eat" and i_2 ="but no" and the analogous phrases d_1 ="the children", d_2 ="sweets" and d_3 ="carrots". The latter correspond positionally to the following phrases from the matched sentence source: δ_1 ="vegetarians", δ_2 ="vegetables", and δ_3 ="meat". The translation of d_1 and d_2 (as we might find it in the lexicon) is "die Kinder" and "Süßigkeiten", respectively. We assume that the lookup of "carrots" yields several alternative translations D'_3, namely "Karotten", "Möhren" and "Rüben". Thus, the substitution generates the following sentences:

(2.3) Die Kinder essen nur Süßigkeiten aber keine Karotten.

(2.4) Die Kinder essen nur Süßigkeiten aber keine Möhren.

(2.5) Die Kinder essen nur Süßigkeiten aber keine Rüben.

In the runtime-approach, the application of the replacement operation is exclusively tailored to input and matched sentences. In the remainder of this section, we describe two strategies of improvement. The first aims at improving translation recall for cases where the input sentence is embedded in the matched sentence. The second aims at avoiding boundary friction through the extension of different phrases.

Matched Sentence Cut

The purpose of the cutting procedure is to improve translation recall for cases in which the input sentence is embedded in the matched sentence. As an example from our corpus, consider the input sentence "Zeker niet." that matches on the example translation: "Nee, zeker niet. → No, certainly not.". Here, the additional phrase "Nee, - No," is cut away such that the remaining "certainly not" can be identified as translation.

[2]Note that d-phrases are the only type of phrases that may match on the empty word ϵ.

```
// input:
input sentence S
minimum number of matches required m
  (e.g. number of words in S divided by 3 plus 1)

//initialisation:
matches = []

//loop through corpus:
For each example in corpus
    min_index = 0
    rule_score = 0
    max_score = m
    //loop through input sentence:
    For each word w in S
        If example source contains w
        Then
            new_index = word index of w in example source
                        after position min_index
            If new_index exists
            Then
                increase rule_score by 1
                min_index = new_index
            End If
        End If
    End For
    If rule_score = max_score
    Then add example to matches
    If rule_score > max_score
    Then
        matches = [ example ]
        max_score = rule_score
    End If
End For
Return matches
```

Figure 2.3: Runtime matching algorithm.

```
//input:
input sentence = w_1 ... w_n
matched example source = s

// initialisations:
p = 0
GAP_I = TRUE // gap in i-phrases
GAP_D = TRUE // gap in d-phrases
i-phrases = [] // list of i-phrases
d-phrases = [] // list of d-phrases

//loop over the input sentence:
For j = 1 ... n
    If w_j occurs in s after position p
        p = end position of w_j in s
        If (GAP_I)
            add w_j to i-phrases
        Else
            extend last i-phrase with w_j
        End if
        GAP_I = FALSE // i-phrase continues
        GAP_D = TRUE  // end of d-phrase
    Else
        If (GAP_D)
            add w_j to d-phrases
        Else
            extend last d-phrase with w_j
        End If
        GAP_I = TRUE  // end of i-phrase
        GAP_D = FALSE // d-phrase continues
    End If
End For
return i-phrases and d-phrases
```

Figure 2.4: Runtime sentence analysis. Divides an input sentence into i-phrases and d-phrases. Note the difference between the methods "add" and "extend": While "add" inserts the new element at the end of the list, "extend" modifies the last element of the list by appending the content of the new element at its end. For the sake of simplicity, the treatment of empty d-phrases is omitted.

```
//input:
matched example source = S
identical phrases = i_1 ... i_m

// initialisations:
p = 0
p' = 0
delta-phrases = []; //collection of delta-phrases

//loop over identical phrases:
For j=1...n
    p' = start position of i_j in S after position (p-1)
    If (p < p')
        delta-phrase = substring of S from position p to p'
        Add delta-phrase to delta-phrases
    End If
End For
return delta-phrases
```

Figure 2.5: Runtime alignment algorithm. Identifies δ-phrases on the basis of positional correspondence to d-phrases.

```
// input:
matched example target: T
d-phrase = d_1...d_n
delta-phrases = delta_1...delta_m

// initialisation:
substitutions = [ T ] // collection of generated translations is
                      // initialised with matched example target

// loop over delta-phrases:
For j = 1 ... m
    delta'-phrases = translations of delta_j
    d'-hat-phrase = translation of d_j that occurs in T
    // loop over delta'-phrases:
    For each delta'-phrase in delta'-phrases
        new-substitutions = [ ]
        // loop over substitutions:
        For each substitution in substitutions
            new-substitution = substitute the first occurrence
                               of delta'-phrase in substitution
                               by d'-hat-phrase
            add substitution to new-substitutions
        End For
    End For
    substitutions = new-substitutions
End For
Return substitutions
```

Figure 2.6: Runtime recombination algorithm: Generates a set of translations by substitution from the translation example target. Note that the actual implementation of this algorithm includes another loop for $\hat{\delta}'$-alternatives, as it does not rely on an unanimous alignment.

Conceptually, "Nee," is a non-empty δ-phrase matching on an empty d-phrase. However, since in the analysis algorithm presented in Figure 2.4 d-phrases are determined on the basis of the input sentence, empty d-phrases at the beginning and end of the input sentence are not detected. Instead, the runtime algorithm will end up with a different number of d- and δ-phrases, and – being unable to align these – will abort the translation process. In order to prevent this, we cut an additional phrase a off the beginning or end of the matched sentence, if the following requirements hold:

1. a is translatable and its translation a' can be identified at the beginning or end of the original target sentence.

2. The borders of both phrases a and a' must be at the start/end of either the sentence or an i-phrase.

Thus, the cutting algorithm is an extension of the sentence analysis algorithm in Figure 2.4, which takes action during the first and last round of the loop over the input sentence. If the above conditions hold, the algorithm modifies the matched sentence source such that additional phrases before w_1 or after w_n are removed from the matched example source. It further cuts off the translation of these phrases from the example translation.

Phrase Enlargement

The aim of phrase enlargement is to reduce boundary friction in the translation. It is based on the assumption that a larger context increases the probability to obtain an accurate translation. Therefore, we extend the analogous d-phrases beyond the borders of their neighbouring i-phrases.

For illustration, consider the following example: Suppose our input sentence "The soup is cold" matches on the translation example "The tea is cold → Der Tee ist kalt". Sentence analysis yields the identical phrases i_1 ="the" and i_2 ="is cold" and the different phrase d_1 ="soup" which is analogous to δ_1 ="tea". This implies the translation rule "The X is cold → Der X is kalt" and would result in the grammatically incorrect translation "Der Suppe ist kalt". Our solution to this problem is to enlarge δ_1 from "tea" to "the tea", and accordingly adapt d_1 to "the soup". Assume the lexicon would contain "der Tee" as translation from "the tea", we obtain a new implicit rule "X is cold.→ X ist kalt.". Provided the lexicon knows the correct translation for "the soup", we then obtain the correct sentence translation: "Die Suppe ist kalt". Thus, phrase enlargement can deal with certain agreement problems in particular, provided that they only affect neighbouring words.

The enlargement algorithm may also extend empty d-phrases. For example, consider the input sentence: "I like this" matched on the translation example "I like this book→ Ich mag dieses Buch" Here, the only d-phrase according to our sentence analysis would be the empty string ϵ. Our sentence analysis yield: i_1= "I like this" , i_2= "." and

$d_1 = \epsilon$. Without phrase enlargement, this results in the translation "Ich mag dieses", corresponding to the translation of i_1. However, if we extend the d_1 to "this", a more appropriate translation, such as "Ich mag es" can be obtained if the lexicon contains the entry "this → es".

The phrase enlargement algorithm consists of the following 3 steps:

1. Enlargement of δ-phrases (see algorithm in Figure 2.7)
2. Adaptation of d-phrases: Appending the pre- and post-phrases of corresponding δ-phrases.
3. Filtering: Rejection of enlargements that yield untranslatable d-phrases.

First, the "largest translatable phrase" $\bar{\delta_i}$ is found for each of the analogous phrases $\delta_i \in \Delta_i$ (see algorithm in Figure 2.7).[3] A δ-phrase is *translatable* if a lookup in the phrase table yields a non-empty set of translations of that phrase, and if at least one of these translations occurs in the example translation

In the current implementation, enlargements are bi-directional but limited to one additional word before and/or after the δ-phrase in the example source (pre- and/or post-phrase). Note that the algorithm maintains the division of pre-phrase, post-phrase and original δ-phrase. This information is needed for the next step, the adaptation of d-phrases. Thus, the d-phrase is enlarged by simply appending the pre- and post-phrases of its aligned δ-phrase left and right to the core d-phrase. During the final filtering phase, all enlargements leading to untranslatable d-phrases (i.e. result of lexicon lookup of the d-phrase returns is empty) are rejected. In sum, an enlargement $\bar{d_i}$ is acceptable if the following requirements are fulfilled:

1. $\bar{\delta_i}$ is translatable.
2. A translation of $\bar{\delta_i}$ occurs in the example translation.
3. $\bar{d_i}$ is translatable.

2.1.3 Compiled EBMT

The compiled approach is an instance of two-phase EBMT. It uses a *precompiled* corpus, which contains a translation grammar in the form of translation rules attached to each translation example. In our implementation, the translation rules consist of *untyped templates*, i.e. they contain some words from the original sentences and variables representing the remainder words. The templates are generated on the basis of the phrase

[3] Notation: δ' is the translation of δ, $\hat{\delta}'$ is the translation that actually occurs in the example, $\bar{\delta}$ is the enlarged δ phrase.

```
//input:
delta-phrases
matched example source S
i-phrases

//initialisation:
enlarged_delta-phrases []
If matched example source starts with i-phrase
Then i-phrase_counter = 0
Else i-phrase_counter = -1

//loop through delta-phrases:
For each delta in delta-phrases
    preceding_i-phrase = epsilon
    subsequent_i-phrase = epsilon
    If i-phrase_counter > -1
    Then preceding_i-phrase = i-phrases[i-phrase_counter]
    If i-phrase_counter < number of i-phrases
    Then subsequent_i-phrase = i-phrases[ i-phrase_counter + 1 ]
    increase i-phrase_counter by 1

    If Not empty (preceding_i-phrase)  AND Not empty (subsequent_i-phrase)
    Then largest_delta =  preceding_i-phrase + delta + subsequent_i-phrase
    If largest_delta is contained in lexicon
    Then
        Add largest_delta to enlarged_delta-phrases
        Continue loop through delta-phrases
    End If

    If Not empty (preceding_i-phrase)
    Then largest_delta =  preceding_i-phrase + delta
    If largest_delta is contained in lexicon
    Then
        Add largest_delta to enlarged_delta-phrases
        Continue loop through delta-phrases
    End If

    If Not empty (subsequent_i-phrase)
    Then largest_delta =  delta + subsequent_i-phrase
    If largest_delta is contained in lexicon
    Then
        Add largest_delta to enlarged_delta-phrases
        Continue loop through delta-phrases
    End If

    Add delta to enlarged_delta-phrases
End For
Return enlarged_delta-phrases
```


Figure 2.7: δ-phrase enlargement algorithm: Generates the "largest translatable" δ-phrases.

Figure 2.8: Sequence diagram for compiled approach.

table: Every occurrence of a phrase table entry phrase is substituted by a variable. A sequence diagram is to be found in Figure 2.8. Due to the pre-compiled translation rules, the overall process during the translation phase is less sophisticated than in the run-time approach. While the transfer phase is simpler, as the rule already contains major alignment information, the matching phase exploits the rule in an additional prefiltering step.

Matching In the compiled approach, matching is performed not only on the sentence pair directly, but primarily on the translation rule. The *similarity metric* is a combination of template-based matching and word-based matching. More specifically, the matching process consists of the following two steps:

1. The input sentence must match on the translation rule source, i.e. it must contain all words of the rule in the right order and may only contain additional words where they match on variables (see Figure 2.9).

2. The number of words that the example sentence has in common with the input sentence must be maximum. From all example translations whose rule matches on the input sentence, we select those with the highest number of matching words (see Figure 2.3).

Transfer The transfer phase of the compiled approach merely consists of instantiating all variables in the rule source. As shown in Figure 2.10, this is achieved on the basis of a comparison between rule and input sentence. Variable sequences lead to several alternative instantiations, as the borders between variable sequences cannot be re-established unambiguously.

Recombination These instantiations are then translated by means of lookup in the lexicon (which typically yields several alternative translations). Finally, the variables in the target of the rule are substituted by these translations. Importantly, all combinations of translation alternatives are generated, as Figure 2.11 shows.

Again, suppose our input sentence is "The children only eat sweets but no carrots." Our corpus contains the same translation example as in Section 2.1.2, only this time it has a translation rule attached, say "X_1 only eat X_2 but no X_3. $\rightarrow$ X_1 essen nur X_2 aber keine X_3.". Then, we instantiate the variables X_1 with "the children", X_2 with "sweets" and X_3 with "carrots". Depending on the entries in the lexicon, the algorithm will yield the same translations as in the runtime approach.

2.2 Evaluation

In this section, we describe our evaluation of the translation system. As corpus, we use a questionary translated from Dutch to English. The corpus is relatively small (4428 sentences), but it contains a lot of similar sentences.

In Section 2.2.1 we show how the system deals with selected real-data examples, before we present a quantitative evaluation of the system in Section 2.2.2.

2.2.1 Qualitative Evaluation

In this section, we discuss selected example translations to illustrate how the different methods work on real data, and to compare the strengths and weaknesses of the different translation methods. We also examine the rule generated from the example itself: If this rule is empty, this means that all phrases of the input sentence are contained in the lexicon.

Example 1: Ja in een gaming room (Yes in a gaming room)

Runtime For this input sentence, the string match only retrieves one translation example:

(2.6) "Ja in een internet café $\rightarrow$ Yes in an Internet café".

Matching against this sample, the input sentence is segmented into one identical phrase i ="Ja in een" and one different phrase d ="gaming room". The latter corresponds positionally to δ ="internet café" in the matched sentence source. The lexicon translates the phrases "internet café" and "gaming room" without changing them. Thus, the runtime approach without use of the *matched sentence cut* or *phrase enlarge* function yields the following incorrect translation: *"Yes in an gaming room".

While, for this translation example, the cut procedure does not have any impact, it is the enlarge function that achieves the correction of the article "an" into "a". It does so by enlarging δ from "internet café" to "een internet café" the translation of which can be found in the phrase table as "an internet café". Accordingly, d is adapted to "een gaming room", the translation of which is "a gaming room". This yields the correct translation "Yes in a gaming room".

```
// input:
input sentence S
//instantiations:
previousTokenIsVariable = FALSE
index_sentence = 0
index_rule = 0
matches = []
abort = FALSE

For each example in corpus
    For each token in source
        If token is Variable
        Then
            previousTokenIsVariable = TRUE
        Else
            index_sentence = first occurrence of token in S
                             after position index_sentence
            If token does not occur in S after position index_sentence
               OR (index_sentence > 0 AND NOT previousTokenIsVariable)
            Then
                abort = TRUE
                Break loop through source
            End If
            previousTokenIsVariable = FALSE
            index_sentence = index_sentence + Length(token) +1
            index_rule = index_rule + Length(token)
        End If
    End For
    If Not abort
    Then
        If previousTokenIsVariable
           OR index_sentence = Length(S) +1
        Then
            Add example to matches
        End If
    End If
End For
Return matches
```

Figure 2.9: Compiled matching: The rule matching algorithm.

```
// input:
input sentence S
translation rule

// initialisation:
instantiations = []
variables = []
current_index = 0
variable_index = 0

// loop through rule source
For Each token in rule source
    If token is a variable
    Then
        add token to variables
        If previous token was non-variable
        Then variable_index = current_index
    Else
        current_index = position of token in S after current_index
        substitute = Substring of S from variable_index to current_index
        For Each possible division of substitute among the variables
            For Each variable in variables
                substitute_part = part of substitute corresponding
                                  to variable in division
                new_rule = variable -> substitute_part
                Insert new_rule into instantiations
            End For
        End For
    End If
End For
If variables is not empty
    substitute = Substring of S from variable_index to current_index
    Add substitution rules to instantiations (see above)
End If
Return instantiations
```

Figure 2.10: Compiled transfer algorithm: Variable instantiation. Variable sequences are instantiated with all possible divisions of the substitute string.

```
//input:
rule target
instantiations

//initialisation:
translations  = [ rule target ]

// loop through instantiations:
For each variable in instantiations
    initialise new_translations as mapping of variables to substitutes
    // loop through alternative substitutes:
    For each substitute associated to the variable
            new_translation  = Replace
                               all occurrences of instantiation variable
                               by instantiation substitute
            add new_translation to new_translations
    End For
    translations = new_translations
End For
Return translations
```

Figure 2.11: Compiled recombination algorithm: Variable substitution.

Compiled In the compiled approach, the correct translation is generated from an empty rule "$X_1 \rightarrow X_1$". This is in line with the fact that the original rule generated from the example itself is also empty. In other words, the sentence can be translated phrase by phrase. However, unfortunately there are a number of additional rule matches on either "in" or "een", which – after long computation times – do not produce a correct translation.

Example 2: Minder dan 1x per maand → Less than 1x per month

Runtime String based matching retrieves the following example translations for this input sentence:

(2.7) "Minder dan 1 keer per maand → Less than once a month"

(2.8) "Minder dan twee keer per maand → Less than twice a month"

(2.9) "Ja minder dan éénmaal per maand → Yes less than once a month"

Processing of the first two examples fails during the transfer phase, as the d-phrases "1 keer" and "twee keer" cannot be aligned. The reason is that these phrases are not contained in the phrase table, because they are not one-on-one alignments ("1 keer → once" and "twee keer → once"). However, example 2.9 can be successfully used by means of the *cut* procedure. The cut procedure reduces the original example translation to "minder dan éénmaal per maand → less than once a month" on the basis of the translation pair "ja – yes". Then, sentence analysis divides the input sentence in the identical phrases i_1 ="minder dan" and i_2 ="per maand", and the different phrase d ="1x" positionally corresponding to δ ="éénmaal". However, the resulting translation "less than 1x a month" does not get the preposition right. This can only be achieved by a combined use of the cut and the enlargement procedure. The latter extends d to "dan 1x per" and adapts δ to "dan éénmaal per", which is translated as "than once a". Since the extended d is now translated as "than 1x per", this results in a correct translation.

Compiled In the compiled approach, the correct translation is retrieved by an empty rule. In fact, the example itself also has an empty rule attached, meaning that all phrases are contained in the lexicon. Apart from this, there is only one more futile rule match. Notably, neither of the translation rules attached to the string matches is of any additional use:

(2.10) "$X_1 \rightarrow X_1$"

(2.11) "Twee keer $X_1 \rightarrow$ Twice X_1"

(2.12) "X_1 1 $X_2 \rightarrow X_1 X_2$"

Example 3: Dit onderzoek duurt ongeveer twee minuten van uw tijd → This survey will take about two minutes of your time

Runtime String match retrieves the following example translation:

(2.13) "Dit onderzoek duurt ongeveer drie minuten van uw tijd → This survey will take about three minutes of your time".

Due to an ambiguity in the translation of the phrase d ="twee" (translated as "twice" or "two"), the runtime approach without phrase enlargement yields two translations:

(2.14) "This survey will take about two minutes of your time"

(2.15) "This survey will take about twice minutes of your time"

However, phrase enlargement can disambiguate, because it prefers the translation "two minutes" of the extended $\bar{d}$ ="twee minuten", after having extended δ ="drie" to "drie minuten" ("three minutes"). Thus, it generates exclusively the correct translation.

Compiled The compiled approach fails completely. The retrieved example rules match on the preposition "van", but the rule's inherent translation of this preposition ("clothing", "for", "on", "taken from the") is in all cases inappropriate in this context. Note: The original rule of the input sentence's example is "corrupted". It has an empty sentence frame on the source side and a non-empty frame on the target side: "$X_1X_2 \rightarrow X_1$ about X_2".

Example 4: Dat is deels gelukt → I was partly successful

Runtime String matching retrieves the two example translations:

(2.16) "Nee dat is niet gelukt → No I was not successful"

(2.17) "Ja dat is gelukt → Yes I was successful"

Both examples require the *cut* procedure, which removes the preceding phrases "Nee – No" and "Ja – Yes", respectively. On the basis of the first example translation, the runtime approach then comes up with the correct translation, but also with the incorrect translation "partly was not successful", due to the mis-alignment of "niet" to "I" in the phrase table. Phrase enlarge can eliminate the incorrect translation alternative, by extending δ ="niet" to $\bar{\delta}$ ="niet gelukt" and such that lookup of "niet" by itself is avoided.

Compiled The compiled approach gets the translation correct using the rule

(2.18) "X_1 dat is $X_2 \rightarrow X_1$ I was X_2"

from the first string match.

Example 5: Sprak mij enigszins aan → Appealed to me somewhat

Runtime The runtime approach generates the correct translation and one acceptable translation ("Did somewhat appeal to me") plus 3 incorrect translations from overall 4 string-matches. The correct translation is generated from the example

(2.19) "Sprak mij persoonlijk aan → Appealed to me somewhat"

Here, phrase enlargement of "enigszins" to "mij enigszins" enables to eliminate the incorrect translation "slightly".

Compiled In the compiled approach, matching on the rule same example gives the correct translation. The rule is:

(2.20) "Spraak X_1 aan → Appealed to X_1"

From the above examples, we can see that in the runtime approach the *phrase enlargement* and the *matched sentence cut* procedures work effectively and even in combination (example 2). Phrase enlargement can either produce a new translation (example 1), or provide a disambiguation by eliminating incorrect translation alternatives (examples 3 to 5). As to the compiled approach, in examples 1 and 2 the correct translation can be retrieved from an empty rule, as all phrases are contained in the lexicon (we can see that from the fact that the rule attached to the input translation example is also empty). In example 1 we further see that our translation rules are not always suitable for matching. A stop word list is necessary to ensure reasonable computation times. The rules retrieved by examples 4 and 5 are particularly nice example rules. Note that Rule 2.18 could not easily be retrieved by any syntax-based method. However, while this rule works perfectly well for the given context, it might produce nonsense translations in a different context. In contrast, Rule 2.20 seems perfectly safe to apply.

In this section we have discussed individual translation examples. In the next section, we evaluate the overall translation performance on the example corpus.

2.2.2 Quantitative Evaluation

We tested our implementation in 10-fold cross-validation on the following corpus: A questionary translated from Dutch to English. The corpus is relatively small and contains many similar sentences. It also contains repetitions of the same source sentences

with contradictory translations. Therefore we do not use exact matches for translation. During preprocessing, we filtered internet addresses and untranslated sentences (instructions for the translator, etc.). We also removed punctuation marks as their usage is rather inconsistent in our corpus.[4]

We compiled a very small list of frequent words to be ignored by the similarity metrics of the compiled as well as the runtime approach.[5] For the compiled approach, we only allow rules to match whose source contains at least one non-variable that is not a frequent word. This is because – due to their high number in our rule set – processing other kinds of rules would slow down the translation system considerably, while there is a low prospect of obtaining a good translation. Instead, we add the *standard rule* "$X \rightarrow X$" to all matched rule sets. We collect the set of all sentences our translation methods return and regard the result as "correct" if the gold standard translation is contained in this set. We do so, because we assume that a monolingual language model will often be able to select the correct translation.

The result of this experiment is shown in Figure 2.12. The result shows that the cut and the enlarge extensions to the runtime approach work effectively. Phrase enlargement can sometimes find new correct translations.[6] The cut procedure is more effective, even though its application is restricted to the requirements presented on page 46. The result further shows that the compiled approach is roughly twice as good as the runtime approach. However, the lexicon for this experiment was collected from the whole corpus. On the one hand, this is in line with our assumption that sufficient word and phrase translations are available. On the other hand, it can also be argued that lexicon entries should not be collected from the test corpus, because this already includes too much information about the translation of the test corpus. Hence we ran the first round of the experiment again, filtering all lexicon entries that do not occur in the training corpus. As shown in Figure 2.13, the performance of the compiled approach decreases to the level of the runtime approach's.

Overall, the result is disappointing, as either method only retrieves about 15% of the gold standard translations. However, given the fact that we have built this translation system from scratch and on a corpus of very small size, a good translation result was not to be expected in the first place. One shortcoming of both approaches is their limited ability to exploit single-word translations from the dictionaries. The runtime operations do evidently improve the translation performance, but at the same time they have the side effect of longer translation times. The compiled approach, on the other hand, is crucially dependent on the quality of the translation rules. Some of our translation rules have been counter-productive, as they contain non-variables exclusively on the target side (e.g. "$X_1 X_2 \rightarrow X_1$ about X_2"). Other rules have been useless for

[4] We consider the following punctuation marks: ".", ",", "!", "?", ":", ";", "(", ")", "/" and "\".

[5] For Dutch the list consists of the following stop words: "van", "ik", "het", "een", "de", "is" "en", "u", "of", "te", "voor", "in".

[6] As shown in Section 2.2.1, it can further discriminate between incorrect and correct translations, but this disambiguation does not show in the result due to our evaluation method.

round	1	2	3	4	5	6	7	8	9	10
Runtime	14.92	16.12	16.72	16.12	14.14	15.27	14.97	17.96	16.77	14.97
+cut	21.79	20.00	19.40	22.39	17.66	20.96	22.46	23.35	20.96	20.96
+enlarge	15.55	16.72	17.61	16.42	14.37	15.57	15.57	18.56	18.26	15.87
+cut & enlarge	21.49	20.00	20.60	22.99	18.26	21.26	22.75	23.65	22.75	21.86
Compiled	31.64	31.05	28.96	29.85	26.05	29.34	31.74	34.43	27.25	29.94

Figure 2.12: Percentage of correct translations in 10-fold-10 cross validation on Questionary corpus without filtering the lexicon.

round	1
Runtime	15.64
+cut	19.87
+enlarge	16.29
+cut & enlarge	20.52
Compiled	14.66
only correct in runtime approach	10.10
only correct in compiled approach	8.79

Figure 2.13: Percentage of correct translations in first round of 10-fold-10 cross validation on Questionary corpus after filtering the lexicon.

matching, because they are no longer representative for the example translation they stem from (e.g. Example 2.11). The best translation performance was achieved by the compiled approach without filtering the lexicon, where the rules have been perfectly complementory to the lexicon entries.

Hence, the crucial question is: What should a translation rule look like? In our view, a translation rule should be two things:

1. Representative for the translation example
2. Adapted to the lexicon and other translation resources

Such a translation rule, we call *translation frame*.

3 Translation Frame Generation

Since the best MT approach seems to consist in a (yet undiscovered) synthesis of SMT and EBMT, our work might contribute to determine specific features of this synthesis. Our focus is on *structural discrepancies*, as we assume the translation of sentences that can be translated in a non-crossing one-to-one aligned fashion to be comparatively trivial.

Our approach relies on the assumption that a sufficient amount of these lexical translations is available. These may be obtained through the combined use of dictionaries and SMT word alignment tools, for example. We believe this assumption is quite realistic, even for *any* language pair. At least, it should be much more realistic than presuming the existence of deep processing linguistic resources (e.g. robust parsers). If available, these linguistic resources (in particular morphological processing) may be incorporated into the system in future work if this improves performance. However, this thesis pursues a language-independent approach based on shallow processing and incorporates a bi-lingual dictionary as the only linguistic resource.

Our method is based upon a *translator-oriented view* of translation, as it particularly emphasises the *compromise* between source sentence and target language that translation always involves.[1] Simply speaking, we assume a translator is torn between the requirements of the source sentence (meaning) and those of the target language (grammaticality). However, more clearly it has been explained by Mansell (2005), whom we would like to mention as our major source of inspiration in this respect. Mansell applies Prince and Smolensky (2002)'s *Optimality Theory (OT)* to the translation of poems. He thereby takes the translator, the "intelligent, thinking" *creator of text*, as the starting point of his theory. In the OT framework, translation competence is modelled by means of universal violable constraints. Importantly, there are two groups of constraints: *faithfulness* constraints and *markedness* constraints. Faithfulness constraints demand fidelity to the input, and markedness constraints demand unmarked output. Notably, while faithfulness constraints "demand a certain relationship between input and output features", markedness constraints: "demand a certain feature in the output, regardless of

[1] "Is translation really possible? [...] The answers we will suggest are: No, in a certain sense translation is impossible [...] Perfect translation is impossible because meanings and interpretations are not like soft and pliant substances extractable from one expression in one language and mouldable without loss or modification into another expression in another language. Languages, on the contrary, are discrete structures, and meanings are inextricably entwined in the structures themselves; the message is enmeshed in the medium. Therefore, during translation, things crack and snap, things disappear, and things are added. The target language is like a Procrustean bed for the source language. The source language is never quite comfortable – it tosses and turns and finds that it can rest its shoulder in this position and its leg in that position, but no position will rest it all at once." (Dyvik, 2005)

whether or not it is present in the input" (Mansell, 2005). Due to frequent contradictions ("faith constraints demand that things stay as they are, and markedness constraints demand change" (Mansell, 2005)), translation involves an inherent compromise between the two types of constraints. The only amendment we make to this translator-oriented view is that we ignore the direction of translation.

In SMT, the compromise between source sentence and target language has been reflected by the two components *translation model* and *language model* (Brown et al., 1990). The ultimate probability of a candidate translation is calculated as the product of two probabilities independently derived from these two components. However, it has been criticised that in this way grammaticality is merely modelled in a *post hoc* fashion rather than built in *at the core* of a translation system (Way & Gough, 2005; Groves & Way, 2005). We propose to model the source-target compromise at the very core of MT: the alignment.

In Section 3.1, we describe an alignment model, which particularly emphasises the compromise between source and target language, because it models dependencies along two dimensions. In Section 3.2, we develop a method how to generate translation rules from two-dimensionally aligned translation examples.

3.1 Two-Dimensional Alignment

In contrast to the translation problem, the task of the *alignment problem* does not involve producing a translation of the input sentence. Rather, it consists of an analysis of the original sentence and its translation, both of which are provided as input. The task is to find a mapping between those words of the two sentences in the different languages that convey the same meaning (Lopez & Resnik, 2006). This mapping may be on a single-word basis (word alignment) or on the basis of contiguous or even discontiguous phrases. Naturally, a reliable alignment provides a good basis for any machine translation algorithm (SMT and EBMT alike). However, when working with state-of-the-art statistical alignment tools, like Giza++ by Och and Ney (2000), it soon becomes clear that the alignment problem can neither be called "trivial" nor "solved" (at least not for small corpus).

From a linguistic perspective, the most unreasonable concept in existing word and phrase alignment models is the notion of *empty cepts*. Naturally, it cannot be assumed that words emerge out of nothing. Rather, what *is* reasonable, is to search for some kind of reason or justification for every word in a sentence pair. In statistical alignment, *justification* is modelled in the form of statistical correlation. However, most existing alignment models have limited their search for justification to the words in the sentence's translation. As illustrated in Figure 3.1, we call the connection between the sentence and its translation the *inter-sentential dimension*. The dimension within the two respective sentences, we call *intra-sentential*. This dimension has been modelled by various grammar theories, but it has (at least to our knowledge) so far been neglected in alignment

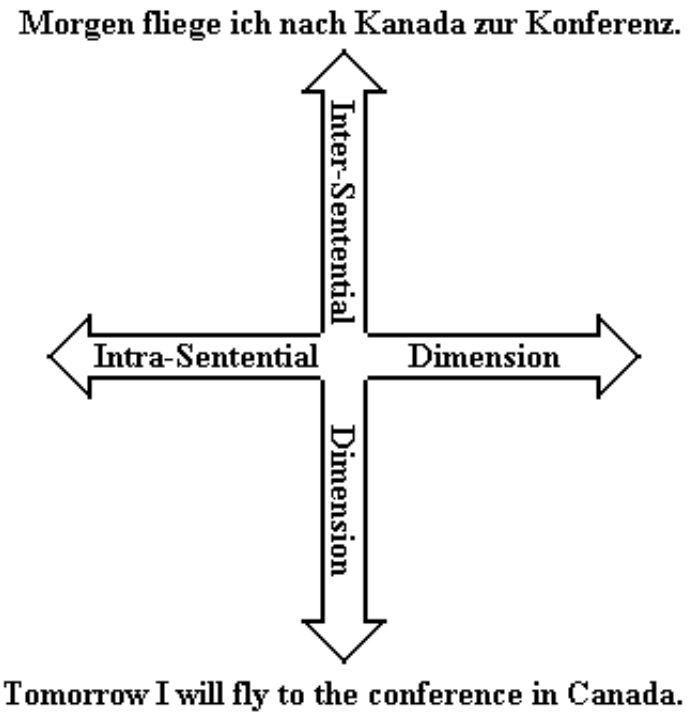

Figure 3.1: Illustration of the intra-sentential and the inter-sentential dimension in translation.

models.

Unfortunately, pinpointing a word's justification within the translated sentence is hard and often impossible, even for humans (see Figure 1.6 on page 14). It must be accepted that not all words can be aligned *directly* via the inter-sentential dimension, because not all words are *translatable* to all languages. Phrase alignment circumvents the problem of finding justifications for single words by treating them as parts of larger chunks (see Figure 1.8 on page 15). Two-dimensional alignment is different, as it models the intra-sentential dimension explicitly in terms of statistical correlations. In other words, it extends the search space along the intra-sentential dimension. It is based on the following assumption: Often it will be more reasonable to search for a word's justification *within the sentence itself*, than in its translation.

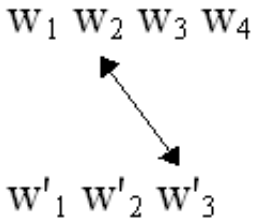

Figure 3.2: Direct alignment of w_2 and w'_3.

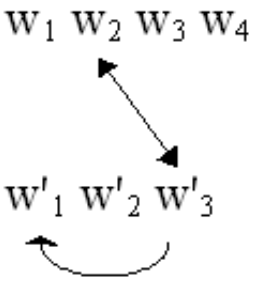

Figure 3.3: Indirect alignment of w'_1.

In order to build up an alignment without empty cept, it is the following question that must be answered for every word w in the sentence pair: Why does w occur in the sentence pair? Along the inter-sentential dimension, the answer should be this: Because it conveys the same meaning as some word w' in the translated sentence. Statistical alignment (Brown et al., 1990) goes one step further as it aims to identify w's translation by means of co-occurrence as main criterion[2]: $argmax_{w'} P(w|w')$. In these terms, we can

[2]To keep things simple, let us ignore distortion and fertility models at this point.

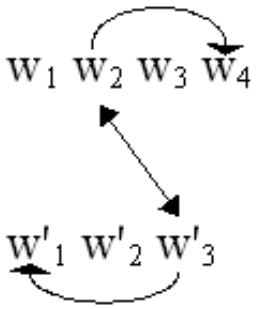

Figure 3.4: Indirect alignment of w'_1 and w_4.

paraphrase the question as: Would w still be part of the sentence pair if it was not for w'? Along the intra-sentential dimension, there is the very same question to be asked. The only difference is that, here, the answer will not yield a translation, but a *trigger* of w. Thus, intra-sentential dependencies can be calculated statistically by maximising the dependent probability $P(w|s)$ of a word w and its trigger s.

Along which dimension a word will be aligned, depends on the strengths of its (competing) inter- and intra-sentential dependencies. Some words can be aligned *directly* via the inter-sentential dimension, because they are *translatable*, i.e. the sentence's translation contains a translation of the word which represents (most of) the word's meaning. For the remaining words, these *direct alignments* (see Figure 3.2) serve as *anchors* to which they can be connected via an *indirect alignment* (see Figures 3.3 and 3.4). In contrast to other word- or phrase-based alignment models, a two-dimensional alignment model can directly model n-to-m non-consecutive word alignments, thanks to the mechanism of indirect alignment. More specifically, alignments containing contiguous as well as discontiguous phrases can be aligned in *two steps*:

1. Establish direct alignment of some of the words by means of *inter-sentential dependencies* (illustrated as bi-directional straight arrows).

2. Indirectly align the remaining words by means of *intra-sentential dependencies* (the curved arrows).

We now illustrate the two-dimensional alignment model on the translation example by Koehn (2004):

(3.1) Morgen fliege ich nach Kanada zur Konferenz. → Tomorrow I will fly to the conference in Canada.

Figure 3.5 shows alignments along the inter-sentential dimension, which we believe can safely be established and that humans can agree on. In contrast, the remaining words

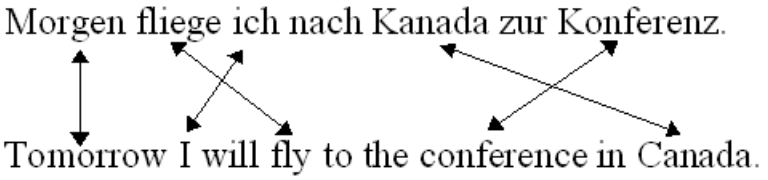

Figure 3.5: Example of inter-sentential dependencies.

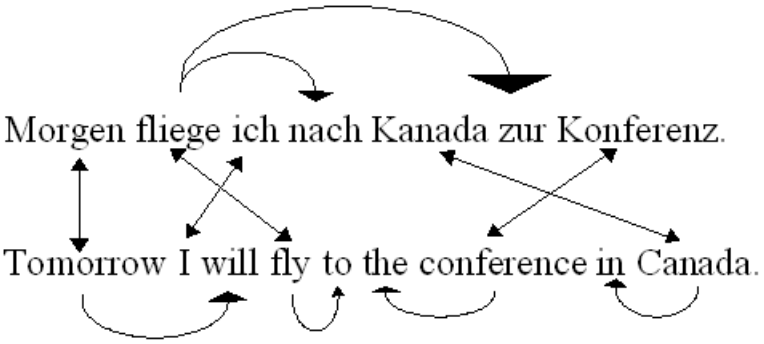

Figure 3.6: Example of intra- and inter-sentential dependencies.

are harder to align directly, and are thus more likely to end up unaligned (i.e. aligned to the empty cept in existing alignment methods, e.g. (Brown et al., 1993)). Strikingly, all the remaining words in this example are function words. We believe this is no accident: Across languages it seems easier to align content words, while function words often lead to ambiguity. One example in this sentence pair is the English word "to": Should it be aligned to "nach" or to "zur"? Humans will not be able to agree on either way, because the meaning of "to" is really distributed over both words. In practice, content words will also appear unalignable along the inter-sentential dimension, but (at least for closely related languages) we expect this being due to sparse data and language evolution more often than to an inherent untranslatability.

We now discuss how the remaining words could be aligned via the intra-sentential dimension. Figure 3.6 shows the intra-sentential dependencies, which we hope can be established for the yet unaligned words in the two sentences. We regard the determiner "the" as depending on the noun "conference", and the preposition "in" as depending on the noun "Canada", because without these words they would not occur in this context, respectively. Note how in grammar theories these particular dependencies may be treated in quite different ways: For example in head-driven grammars, it is noun and preposition that are the head of a noun and prepositional phrase, respectively. As to the preposition

"to", we regard it as depending on the verb "fly", because it can only be used with verbs of movement ("fly", "go", etc.) but not possibly with a verb like "sit". Finally, we would like to point out the dependence of "will" on "Tomorrow", because this yields the following discontiguous phrase translation: "Morgen" → "Tomorrow ... will". However unlikely that such a phrase will be covered by grammar theories, it appears particularly useful for the purpose of MT. Finally, note that with regard to contiguous phrases, such intra-sentential dependencies have the same advantage as discontiguous phrases even if the words are adjacent: Additional words can be inserted in between without this causing any additional problems. For example, an adjective could be inserted between "the" and "conference" (e.g. "the annual conference") without the intra-sentential dependency being affected.

Before we discuss other examples of two-dimensional alignment, we must admit that all dependencies we indicate in the examples of this chapter are merely based on our own intuition and hence cannot be taken for granted. We propose a conservative use of inter-sentential dependencies leaving all controversial alignments to the intra-sentential dimension. Hence, a certain degree of ambiguity in the latter dimension is unavoidable. Since we establish intra-sentential dependencies by statistical methods, the actual alignments will decisively depend on the underlying corpus. The example's main purpose is to show that two-dimensional alignment offers enough flexibility to cover a wide range of structural discrepancy phenomena.

Let us now consider an example translation from French to German (English translation: "He looks at the painting."):

(3.2) Il regarde le tableau. → Er schaut das Bild an.

The German verb "anschauen" (English: observe, look at) has a separable prefix "an" which (depending on sentence structure) can be placed quite far away from the remainder verb. In conventional word alignment, "an" would most likely be aligned to the empty cept, because the source word "regarde" can be aligned to at most one target word, and this will probably be "schaut". However, as shown in Figure 3.7, our model can align "an" indirectly to "regarde", via the intra-sentential dependency of "an" on "schaut".

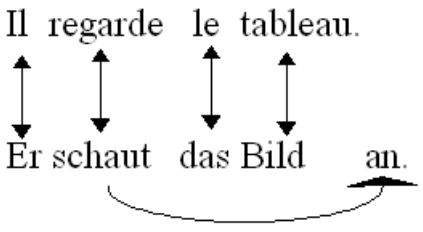

Figure 3.7: Indirect alignment of a separable prefix in a French-German translation example.

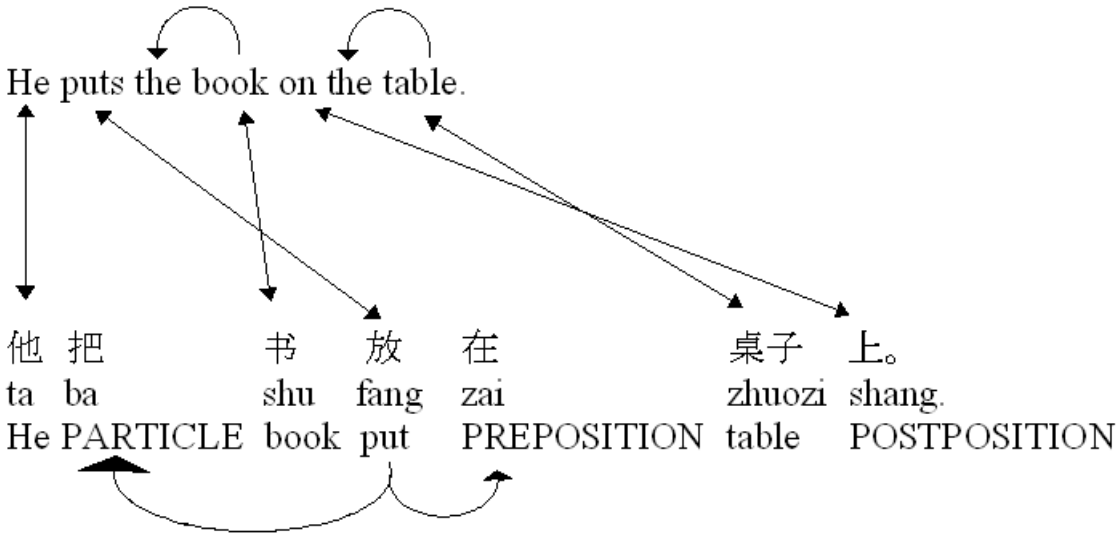

Figure 3.8: Alignment for English-Chinese (Mandarin) translation example.

Due to its inherent flexibility, we believe that the two-dimenisonal alignment model may be particularly suited for translation between very different (unrelated) languages. Figure 3.8 shows a translation example in which an English sentence is translated into a Chinese *bă-sentence*. The Chinese particle 把 bă does not actually have any meaning, except that bă-sentences emphasise the result or effect of the sentence's action – which must be directed to someone or something – more than plain sentences. More importantly, 把 bă changes the sentence's word order from standard SVO to SOV, more specifically: Agent (Subject) + ba + Object + Verb. Since bă is untranslatable, it must be aligned indirectly. In (Schuch, 2008), we have conducted experiments in which we extracted bă-triggers from Chinese sentences, namely the word s which maximises the probability $P(bă|s)$. The result showed that bă is more often triggered by verbs than by any other word class. In particular the verb 放 fàng is a frequent bă-trigger: In experiments with the PH corpus (newspaper text) it was the third-most frequently extracted bă-trigger (extracted 71 times). Hence, an intra-sentential dependency of bă on 放 fàng is most likely. Just as 把 bă is untranslatable into English, the definite determiner "the" is by itself untranslatable into Chinese. In English, however, some determiner is required by the nouns "book" and "table" in this context. Finally, we would like to point out the different treatment of the English preposition "on" and the Chinese preposition 在 zài: If we look at it closely, the English "on" has a function (the function of a preposition) as well as a meaning (namely "on top of" rather than e.g. "underneath"). In Chinese, these two are realised separately, the function by 在 zài and the meaning by the postposition

上 shàng [3]. Thus, we can easily align "on" directly to 上 shàng, while 在 zài (despite being of the same word category) can only be aligned indirectly. Where exactly this word will be aligned, however, will depend on the details of the dependency-calculation method.

Two-dimensional alignment can be implemented with purely statistical methods. This has been shown by Fraser and Marcu (2007), whose LEAF model is a generative alignment model, which directly models M-to-N non-consecutive word alignments via the intra-sentential dimension. Statistical modelling profits from that fact that dependencies are modelled between single words rather than combinations of words, because single words are more likely to occur in corpora with representative frequency. At the same time, two-dimensional alignment seems more plausible for human annotators, which means that better quality training data can be produced by human annoation. However, a statistical model might not be the only way to establish two-dimensional alignment. One particular advantage of two-dimensional alignment is that intra-sentential and inter-sentential dependencies can be established *independently* of each other. Theoretically, both types of dependency can be established by means of statistical alignment, linguistic resources or a combination of both. In practice, the availability of vast amounts of monolingual corpora (in contrast to bilingual corpora, which are still comparatively scarce) may allow to build a reliable statistical model of the intra-sentential dimension, while the inter-sentential dimension might profit from the availability of bilingual dictionaries. That way, the two-dimensional alignment model could also incorporate contiguous and even discontiguous phrases as basic building blocks, given their availability in the lexicon. On the inter-sentential dimension, these could capture the meaning of phrases that cannot be decomposed into individual words (c.f. idioms). On the intra-sentential level, this could capture grammatical relations between phrases rather than words (which exist in most grammar theories). However, in this thesis, we will restrict ourselves to words as building blocks of our alignment model.

So far, the LEAF model is the only implementation of two-dimensional alignment we are aware of. Fraser and Marcu (2007) have developed an unsupervised as well as a semi-supervised generative model, and have shown them to be effective for translation

[3] c.f. 他 tā 把 bǎ 书 shú 放 fàng 在 zài 桌子 zhuǒzi 下 xià 。 – He puts the book under the table.

within an SMT system.[4] Fraser and Marcu (2007) classify the words on the source side as head words, non-head words, and deleted words; and the words on the target side as head words, non-head words, and spurious words.

> The purpose of head words is to try to provide a robust representation of the semantic features necessary to determine translational correspondence. This is similar to the use of syntactic head words in statistical parsers to provide a robust representation of the syntactic features of a parse sub-tree. (Fraser & Marcu, 2007)

In LEAF, a head word is linked to zero or more non-head words; and each nonhead word is linked to from exactly one head word. Fraser and Marcu (2007)'s links between head words of the source and target sentence are what we call inter-sentential dependencies. Their links between a non-head word and a head word are our intra-sentential dependencies. LEAF has some deficiency, as part of the probability mass in the model is allocated towards infeasable alignment structures concerning non-spurious target word placement and source word linking. However, in contrast to other alignment models, it does not have any structural restrictions such as 1-to-1, 1-to-N or phrase-based (all of which can be implemented as special cases of LEAF).

For an EBMT system, we believe LEAF is one possible implementation of two-dimensional alignment – but not necessarily the only way. LEAF's generative story consists of the following steps:

1. determine the word type of each source word
2. find a source head word for each non-head source word
3. determine a target word to be linked to each source head word
4. determine the number of non-head target words to be linked to each target head word
5. determine the number of spurious words

[4] For this, Fraser and Marcu (2007) implemented the following local search operations:

1. move French non-head word to new head
2. move English non-head word to new head
3. swap heads of two French non-head words
4. swap heads of two English non-head words
5. swap English head word links of two French head words
6. link English word to French word making new head words
7. unlink English and French head words.

6. determine spurious words
7. link non-head target words to each target head word
8. determine the positions of target words (head and non-head)
9. determine the positions of all spurious words

The modelling of intra- and inter-sentential alignments is covered by steps 2, 3, and 7. However, LEAF goes beyond this, as it also attempts to predict the number of target words and their positions in the target sentence. Within an SMT system, this is necessary, because the alignment model is used for generating new translations. Thus, LEAF attempts to predict how many non-head target words will be linked to every head target word t (step 4) on the basis of the identitiy of t and the cept size of its inter-sententially aligned source head word. In step 1, it does not only distinguish head words from non-head words, but also attemps to predict which source words s will be deleted in the target sentence, on the basis of the identity of s. Likewise, it attempts to predict "spontaneously appearing" spurious words s on the basis of the identity of s (step 5). From a linguistic perspective, not all of LEAF's predictions seem feasible, given that they are based on so little information. However, within an EBMT system, the alignment model can focus exclusively on the analysis of existing translation examples, if the task of generating new translation examples is taken over by translation rules. As a consequence, only steps 2, 3 and 7 must be modeled explicitely by alignment. All other steps should be regarded as superfluous (in particular steps 8 and 9), or a means to an end (for example, step 1 may help to increase alignment precision). For two-phase EBMT, it is sufficient if their corresponding information appears implicitely in the translation rule.

In the next section, we discuss how to generate translation rules from two-dimensionally aligned translation examples.

3.2 Translation Frames

While in Section 3.1 we have argued in favour of a two-dimensional alignment model, in this section we discuss how such an alignment could be used for *preprocessing* in two-phase EBMT.

As discussed in Chapter 1, preprocessing has the advantage that it can relieve the computational load during translation. This is the case if the resulting data structure

1. is faster to process than the plain string
2. provides additional information needed for translation.

A frequently used data structure are *templates*, in which some parts on both sides of the translation example are replaced by variables. Templates introduce the following mechanisms to avoid boundary friction:

1. If the template is representative of the original sentences, *template matching* allows for a more informed similarity metric than matching on the overall source sentence.

2. Words that occur in the template are *excluded from replacement*.

3. *Typed variables* allow for the *unified treatment* of various kinds of translation knowledge (Kaji et al., 1992).

4. Templates may contain *word re-ordering* information implicitely, i.e. they do not depend on an explicit (distortion) model.

5. Templates may introduce *additional words on the target side* without linking them to an empty cept (rather, they are linked to the overall template).

Preprocessing consists in some form of *generalisation* of the original example translations in the corpus. However, this should be regarded as a controversial issue, because it bears the risk of losing crucial information:

> "[Since] variables in templates allow for paradigmatic variations at some predefined positions only [...] templates may well be insufficient in representing all of the implicit knowledge contained in examples" (Lepage & Denoual, 2005b).

For example, consider the template "*X salts Y*" generated from the sentence sentence:

(3.3) "The butcher salts the slice."[5]

> "Firstly, it prevents the butcher from being changed into a plural: the butchers. Moreover, it overlooks the fact that salts may also commute with its past and future forms, etc.: salted, will salt, etc., or with cuts, smokes, etc.; and so forth." (Lepage & Denoual, 2005b)

This example illustrates what we call the *preprocessing dilemma*:

1. Preprocessing must not delete information needed for translation of an unseen input sentence.

2. Preprocessing must generalise sufficiently to match on an unseen input sentence.

These two requirements are in conflict with one another.

In our view, these conflicting requirements can only be reconciled by a very *sensitive* preprocessing method that minimises the risk of information loss. Everything depends on whether we can make a distinction between important and unimportant information during the preprocessing phase. The biggest obstacle consists in the fact that during

[5]The example originally stems from (Carl, 1998).

preprocessing we have no knowledge of the sentence to be translated. Given that the input sentence is unknown, the best we can do is evaluate statistically which part of the information is most *likely* to be helpful or unhelpful for translation. However, other 2-phase EBMT approaches (Carl, 2001; Cicekli & Güvenir, 2001; McTait, 2001) do not introduce such a likelihood factor, and thus forego the possibility of automated tuning.

Our aim is to generate templates which are *representative* of the sentence pair. Such a template, we call *translation frame*. In order to be representative, the translation frame must maintain all translation-relevant information that cannot be retrieved from other information sources available during translation. All other information should be abstracted away in order to maximise the chance of matching on an unseen input sentence. During preprocessing, we have no knowledge of the input sentence to be translated. However, we still have the following information:

1. Knowledge about source and target language
2. Knowledge about translation resources

Source and target language knowledge is telling us what kind of source and target sentences, respectively, we can expect as a likely translation task. The translation resources determine which part of the information contained in the full string example can be re-established if required during the translation phase. This part of the information should be abstracted away. For example, in contrast to the templates extracted by Cicekli and Güvenir (2001) and McTait (2001), translation frames shall not contain single word or contiguous phrase translations, because those are better stored and retrieved from a lexicon. As to language knowledge, there are two sources: the (statistical) properties of a corpus, and linguistic theory.[6] As to translation resources, our approach requires a bilingual dictionary at the minimum.[7] Further information, e.g. POS, may be incorporated if available.

With a lexicon as main translation resource, the most important task of the translation frame is to capture the *structural discrepancies* between source and target sentence. Structural discrepancy, in our view, is any divergence from *non-crossing word-by-word* (or phrase-by-phrase) alignment. From this point of view, there are two indicators for a structural discrepancy:

1. The occurrence of a word whose translation cannot be found in the other sentence (*untranslatability*)
2. A *divergence of word order* between source and target sentence

[6] While linguistic theory can deal with higher levels of complexity because it does not suffer from a sparse data problem, corpus information is more easily computed and verified. Thus, the best result can be expected by combing these two resources.

[7] In relation to other linguistic resources, such as parsers or even just POS taggers, we believe that assuming the existence of a bilingual dictionary is relatively realistic for any language pair.

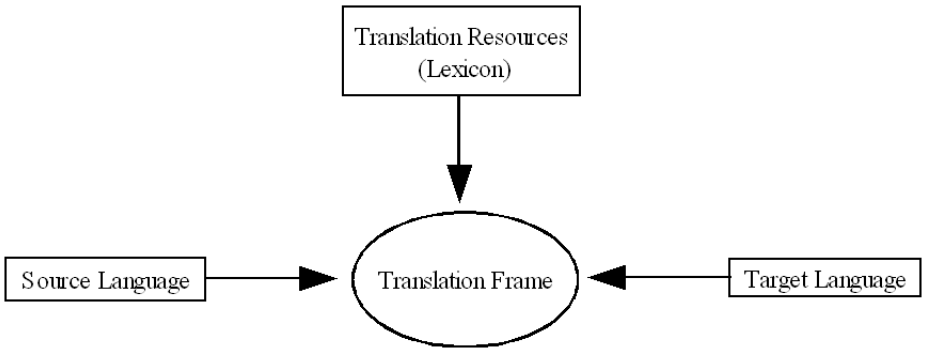

Figure 3.9: Major influences on the translation frame.

A translation frame that is supposed to capture these structural discrepancies is *more* than a mere combination of two sentence frames. As illustrated in Figure 3.9, the translation frame is influenced by the lexicon, as well as the source and target language. In order to combine the latter two, source and target frame must be constructed *interdependently* on one another. In the following section, we discuss how to capture untranslatability in a translation frame on the basis of the two-dimensional alignment model from Section 3.1.

How to Capture Untranslatability in a Translation Frame

We divide the words in the example sentences into two categories:

translatable words: Pairs of words in both sentences that can be aligned directly by means of the language resources (lexicon).

untranslatable words: Words in the sentences that cannot be aligned directly (either because a translation of the word does not exist in the other language, or because it does not occur in the sentence pair).

In the translation frame, these two word types should be treated differently: Normally, translatable words are good candidates for being generalised over in a translation frame. This is because the alignment of translatable words can always be re-established during the translation phase as required. (This will be necessary in case the input sentence also contains the same translatable word). In contrast, untranslatable words should be included in the translation frame, because there is no way to re-establish them during the translation phase.

However, abstracting over all translatable words often will not yield a representative translation frame, because the meaning and usage of untranslatable words is typically

dependent on certain other words they cooccur with. For example, let us reconsider translation example 3.2 on page 71. The separable prefix "an" is the only untranslatable word in this example. However, if we merely include "an" into the translation frame, we obtain the following translation frame:

(3.4) $X \rightarrow X$ an.

This translation frame is counter-productive, because it matches on any possible sentence and then produces an additional word ("an") out of thin air. Thus, building the translation frame exclusively out of untranslatable words is clearly insufficient. Rather, the translation frame must also include (information about) additional words in *support* of these untranslatable words. In traditional alignment models, untranslatable words like "an" are aligned to the *empty cept*. As we have seen in the example above, empty cepts pose a particular threat to translation frames.

In contrast to traditional word alignment models, the *two-dimensional alignment model* does not align untranslatable words to the empty cept. Rather, untranslatable words are aligned indirectly via intra-sentential dependencies to support words. Thus, the two-dimensional alignment model introduces an easy option to generate translation frames which cover all occurrences of untranslatability, namely:

1. Include all words into the translation frame
   a) which take part in an indirect alignment,
   b) or which cannot be aligned at all (if there are any).
2. Replace all other words by variables.

This way, alignment to the empty cept as in SMT alignment models, including LEAF, is avoided completely. Just like Fraser and Marcu (2007), we align untranslatable words indirectly whenever possible. However, even if it is not possible to establish an intra-sentential dependency for it, the untranslatable word will not end up aligned to the empty cept. Rather, it will be aligned to the overall translation frame.

If we apply this method on the two-dimensional alignment of example 3.2 (Figure 3.7 on page 71) the following translation frame is generated:

(3.5) "X regarde Y. $\rightarrow$ X schaut Y an."

This translation frame provides a good basis for translating sentences such as:

(3.6) La fille regarde le ciel. $\rightarrow$ Das Mädchen schaut den Himmel an. (The girl looks at the sky.)

Note that both parts of the translation frame (source and target side) are dependent on the source as well as the target sentence. For example, consider this alternative translation of the French sentence to German:

(3.7) Il regarde le tableau. → Er betrachtet das Bild.

In contrast to example 3.2, this alternative does not contain any untranslatable words, and is aligned in a non-crossing 1-on-1 manner. Without any structural discrepancies needing to be represented, the translation frame of example 3.7 remains empty. Likewise for other target languages, the very same source sentence can be turned into different source frames, according to the particular structural discrepancies between the languages. In contrast, if source and target frame are generated independently of each other, the source frame is the same for all target translations and target languages. In our view, *interdependence between source and target frame* is a prerequisite for capturing structural discrepancies between source and target sentence.

Let us next take a look at a translation example between very remotely related languages, the alignment from Figure 3.8 on page 72 (English–Chinese):

(3.8) "*W* puts the *X Y* the *Z* .→ *W* 把 bǎ *X* 放 fàng 在 zài *Z Y* 。

This translation frames can be used to produce new translations of unseen input sentences, for example:

(3.9) She puts the box under the bed. → 她 tā 把 bǎ 箱 xiāng 放 fàng 在 zài 床 chuáng 下 xià 。

Our method can provide enough flexibility to cover structural discrepancies of even remotely related languages because – in contrast to word-based SMT – there is no need to restrict the search space. The above translation example also shows that POS on its own is not always a reliable indicator for whether or not a particular word should be generalised over. For example, one might assume that prepositions should be included in a translation frame, because in most cases they are hardly translatable to another language. However, based on this syntactic assumption, the matching opportunity "on the bed"→ "under the bed" would be lost. In fact, "on" and "under" are not pure function words in this context, because they do convey a certain meaning – and this meaning is translatable between English and Mandarin Chinese. Thus, while we might hope that our translation frames may bear some resemblence to Cowan et al. (2006)'s extended projections (see Section 1.1.3), we are hesistant to impose any syntactic pre-assumptions. That way, our method – if it was successful – could provide a test bed for syntactic assumptions. We believe it is necessary to double-check syntactic assumptions, in particular for remotely related language pairs. This is because syntax is a monolingual description of language, and not a description of the structural discrepancies between two languages.

Our translation frame generation method is a sensitive form of pre-processing, as it is acting conservatively in order to avoid information loss. In particular, if a translation example contains too many structural discrepancies, the method will refrain from any generalisation and simply return the example string unprocessed. Such is the case with Figure 3.6 on page 70. However, let us consider a slightly simpler sentence pair:

(3.10) Morgen fliege ich nach Kanada. → Tomorrow I will fly to Canada.

Figures 3.10 and 3.11 show two alternative alignment models for this sentence pair, the latter being based on the assumption that the inter-sentential connection between English "to" and German "nach" can be established. Based on those two alignments, our method would generate two different translation frames:

Frame for Figure 3.10: Morgen fliege X nach Y . → Tomorrow X will fly to Y."

Frame for Figure 3.11: Morgen X Y Z. → Tomorrow Y will X Z."

This shows how the lexicon exerts influence on the translation frame in our method. Both translation frames incorporate a translation phenomenon that seems rather difficult to capture by means of syntactic modelling: The change of tense from present tense (German) to future (English). However, these examples also reveal that our translation frames are faced with overgeneralisation. Variable sequences, as in the example above, are particularly prone to overgeneralisation as they match on very flexible phrase length (one phrase shorter, the other longer). Fortunately, overgeneralisation can be reduced by POS typed variables (Cicekli, 2005), e.g.:

Figure 3.11 (typed): Morgen X_{verb} $Y_{pronoun}$ $Z_{preposition}$ W_{noun}. → Tomorrow $Y_{pronoun}$ will X_{verb} $Z_{preposition}$ W_{noun}."

Figure 3.10: Alignment example without lexicon entry "to–nach".

Figure 3.11: Alignment example including lexicon entry "to–nach".

To sum up, on the basis of two-dimensional alignment, we can generate translation frames that are sensitive to source and target language as well as to the lexicon, just as depicted in Figure 3.9. Moreover, the source and target side of our translation frames are interdependent on one another. Our translation frames seem flexible enough to capture structural discrepancies even between remotely related language pairs. We have also provided some linguistic examples to show that they could be useful for translation. In the next section we examine their usefulness with respect to real data.

3.3 A Prototype Implementation

We have implemented a prototype system, which puts into practice the most fundamental ideas of our approach. In order to keep things simple, we did not include any linguistic information and exclusively rely on statistical measures, which are easy to compute. We used the compiled translation method, the corpus and the lexicon from Chapter 2.

Our translation frame generation method exclusively uses single word translations, which are provided in abundance by the dictionary component of our lexicon. Further single word translations are provided by the phrase table, using the *sub-phrases lookup* function described Section 2.1.1. Since the compiled translation method is based on phrase translations the overall translation process imposes a new demand on the lexicon: For non-crossing 1-on-1 aligned phrases the translation of single words must be *consistent* with the phrase translations. More specifically, the lexicon must enable the lookup of every single word as well as the whole phrase (and yield the same result). While the translation table fulfills this requirement, the dictionary does not because it does not have a phrase lookup mechanism. As a consequence, our prototype has one major shortcoming: The compiled translation algorithm from Section 2.1.3 can only take advantage of a translation frame if every variable matches on a single word. We believe the shortcoming can be overcome in either of the following two ways:

1. either by a more efficient implementation phrase lookup in the dictionaries
2. or by adapting the compiled translation algorithm such that it can handle ϵ-matching on variables.

We describe the prototype's alignment method in Section 3.3.1, the translation frame generation algorithm in Section 3.3.2, and a real corpus example in Section 3.3.3.

3.3.1 Calculating Intra- and Inter-Sentential Dependencies

In this section, we describe how we determine intra- and inter-sentential dependencies.

Inter-sentential dependencies are determined on the basis of the lexicon. The condition for an inter-sentential dependency of a word t in the target sentence on a word s

in the source sentence is merely that the lexicon lookup for s contains t. Unfortunately, the lexicon offers no way to determine the *strength* of an inter-sentential dependency. Accordingly, there may be several inter-sentential dependencies for one s or one t.

For intrasentential relationships, we use a *language model* that abstracts over the order and positions of the individual words in the sentences, hence regarding a sentence as an *unordered set of words*.[8] Thus, the conditional probability $P(w_1|w_2)$ signifies the probability of a the word w_1 occurring in a sentence where w_2 occurs (regardless of their positions). Given the definition of conditional probability for two events A and B

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

we determine its value by counting occurrences $P(w_2)$ and co-occurrences $P(w_1 \cap w_2)$ of the words w_1 and w_2 within sentences.[9] The correlation value $P(w_1|w_2)$ determines the *strength* of the intrasentential relationship from the word w_2 to the word w_1 (directed relationship). It ensures that low-frequency words w_2 can be good triggers of w_1. This liberal language model, we have already used successfully to determine triggers of the Chinese *bǎ*-particle (Schuch, 2008).

3.3.2 Translation Frame Generation Algorithm

Since our prototype system is limited to untyped templates, the template generation problem essentially consists of the following question: Which of the words in the example should be included in the translation frame and which should be replaced by variables?

The basic idea of our algorithm is the assumption that every word w in the source and every word w' in the target sentence must be "justified" either by an inter-sentential or an intra-sentential dependency:

1. Either w is a translation of a word w' in the sentence of the other language S' (inter-sentential dependency),
2. Or w is triggered by another word s within the same sentence S (intra-sentential dependency).

Our *translation frame generation algorithm* includes the following words of the source as well as the target sentence into the translation frame:

untranslated words u: All words u in the sentence S that cannot be found to be a translation of any of the words w' in the other[10] sentence S' (according to lexicon lookup).

[8] Note that we can ignore word order in the language model, because the rules are actually taking it into account as an additional information source.

[9] Since our language model is much bigger than a bi-gram model, we first determine all words w_1 in the corpus, for which the calculation of $P(w_1|w_2)$ is needed (because they are "untranslated"), and then only calculate those counts – thus avoiding unnecessary storage use.

[10] We used the term *other sentence* to indicate

supporting words s**:** For every untranslated word $u \in S$, the supporting word s is the (translatable) word in the same[11] sentence with the highest correlation value $P(u|s)$.

translations s'**:** All words $s' \in S'$ that are translations of a supporting word $s \in S$.

The remaining words T are all directly aligned to their translations, and are not involved in any indirect alignements. They will be replaced by a variable, if their alignment is non-ambiguous.

For every word $v \in T$ to be replaced by a variable, there are two sources of ambiguity:

1. The other sentence might contain more than one translation of v.
2. The translation of v might at the same time be a translation of a different word $t \in T$.

In order to enhance the chance of a word being replaced by a variable, we have implemented one *disambiguation heuristic* that gives preference to those v, which can only be aligned to exactly one v' (in source-target as well as target-source direction). More specifically, we align two variables v and v', if at least either of them has exactly one alignment. You find the variable disambiguation algorithm in Figure 3.13 and an example in Figure 3.12. Afterwards, the variable alignment algorithm presented in Figure 3.14 ensures that only non-ambiguous alignments will be established. All words t, whose alignment cannot be disambiguated by our heuristics are included in the sentence frame.

Finally, we have implemented a *variable sequence* clean-up algorithm, which removes variable sequences that are stable in the example with respect to translation. The algorithm first identifies variable sequences that occcur on the source as well as on the target side of the translation example. It then replaces such variable sequences by the first variable of the sequences. It is important to do this step, because otherwise the matching would be restricted to sentences that have exactly the same number of words as the example translation. For example, consider the sentence

(3.11) I like this book. → Ich mag dieses Buch.

It might yield the rule

- the target sentence, if S is the source sentence
- or the source sentence, if S ist the target sentence.

[11] We used the term *same sentence* to indicate

- the source sentence, if S is the source sentence
- or the target sentence, if S ist the target sentence.

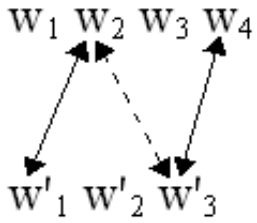

Figure 3.12: The variable disambiguation algorithm filters out the dashed alignment. The other two alignments are established: the left one because w'_1 is exclusively aligned to w_2, and the right one because w_4 is exclusively aligned to w'_3.

(3.12) I like X Y. → Ich mag X Y.

The algorithm identifies the variable sequence X Y and changes it to X:

(3.13) I like X. → Ich mag X.

such that the rule also matches on sentences like

(3.14) I like books. → Ich mag Bücher.

The overall rule generation programme has the following parameters (to be set by the user before running it):

1. Threshold for $P(u|s)$, the minimum probability for an intra-sentential alignment to be established
2. Maximum number of support words per untranslatable word
3. Support words must be translatable (or not)

In the next section, we show how the algorithm deals with an example translation taken from our corpus.

3.3.3 A Real Corpus Example

With the translation system from Chapter 2, we have run another 10-fold cross-validation as described in Section 2.2.2. This time, the compiled translation algorithm (see Section 2.1.3) uses the rules generated by our translation frame generation prototype. Since the size of the corpus is very small, all probabilities are computed on the full corpus. In addition, we experimented with Good Turing Smoothing, but this did not influence

```
// input:
   source_word
   source_sentence
   target_sentence

n = number of translations of source_word in target_sentence
If ( n == 1 )
Then establish alignment between source_word and target_word
Else
   For Each target_word in target_sentence that is a translation of
                                                          source_word
        m = number of translations of target_word in source_sentence
        If (m == 1)
        Then establish alignment between source_word and target_word
        End If
   End For
End If
```

Figure 3.13: Variable disambiguation algorithm.

```
// input:
alignment a
s = index of source_word in source_sentence
t = index of target_word in target_sentence

If s is already aligned in a Then Return Error

If t is not yet aligned in a
Then align s and t
Else remove all alignments of t
End If

Return a
```

Figure 3.14: Variable alignment algorithm.

Round	1	2	3	4	5	6	7	8	9	10
Baseline Rules	31.64	31.05	28.96	29.85	26.05	29.34	31.74	34.43	27.25	29.94
Translation Frames	31.34	30.75	28.06	30.75	25.45	30.84	31.44	33.53	27.84	32.04

Figure 3.15: Evaluation on Questionary corpus (unfiltered lexicon).

the result.[12] As to the programme parameters, we set them in a most liberal way, because other settings did not seem to have a positive influence on the translation result: We allow the programme to use all support words with $P(u|s) \geq 0.5$ (translatable and untranslatable support words alike).

As shown in Figure 3.15, the overall translation performance is almost identical to that of the baseline rules. We suspect that the translation frames cannot perform any better mainly because our dictionary cannot handle phrase translations. During the preprocessing phase, the prototype replaces phrases by variables by looking up every word individually first, and only then turning them into phrases (variable cleanup). However, during the translation phase, the compiled EBMT algorithm attempts a lookup of the whole phrase – and cannot re-establish the translation. In other words, input sentences matching on a variable with more than one of its words cannot benefit from the additional single-word translations provided by the dictionary. As stated earlier, we believe this shortcoming of our prototype can be overcome by a different implementation of either the lexicon or the compiled translation method.

We now present a translation frame generation example from the Questionary corpus that has not been affected by this problem. Here is the translation example from which the rule will be generated:

(3.15) "Kunt u van de tweede persoon in uw huishouden aangeven in welk jaar hij of zij geboren is → Please indicate the year of birth of the second person in your household"

Note that the example contains structural discrepancies, despite Dutch and English being a rather closely related language pair. Let us consider a more literal translation of the Dutch sentence:

(3.16) "Can you indicate of the second person in which year he or she was born".

The "literal" translation still contains a structural discrepancy, as changing the position of the verb "aangeven/indicate" is required to produce a grammatically correct English sentence. It has probably been dismissed by the translator due to its low quality (too

[12] The smaller a corpus is, the more it can be expected to suffer from the *sparse data problem*: Natural language generally contains a high number of different low-frequency words, and low number of different high-frequency words (roughly corresponding to *Zipf law*). Since moreover, language evolution constantly produces new words, there is a considerable amount of word combinations that cannot be observed in the training data (*unseen events*). We perform *Good-Turing smoothing* on the word cooccurrences. Good-Turing smoothing is a *discounting* method which redistributes part of the probability mass from the seen to the unseen events. This is achieved by adapting the frequencies of the seen events according to the following formula:

$$r^* = (r+1) \times \frac{n_{r+1}}{n_r}$$

with r being the frequency ("rank") and n_r the number of events with frequency r.

impolite, too awkward). We believe it would be highly complex to capture the actual translation from the corpus in a syntax-based translation rule.

Our algorithm first determines *support words* s for all *untranslatable words* u in source and target sentence, and then includes these together with their translations into the sentence frame. On the source side, the untranslatable words are: "kunt", "welk", "hij", "zij", "geboren" and "is". Figure 3.16 shows the smoothed cooccurrence probabilities $P(u|s)$ above 0.5 for these untranslatable words u their support word s from the source sentence. On the target side, there are two untranslatable words: "birth" and "please". While there is no support word for "birth" at a probability above 0.5, "please" has the following supports: "birth" (0.81), "indicate" (0.81), "person" (0.71), and "household" (0.65).

u	s ($P(u\|s)$)
kunt	aangeven (0.91), persoon (0.88), hij (0.74), zij (0.67), geboren (0.67), huishouden (0.63)
welk	–
hij	zij (0.67), geboren (0.67)
zij	hij (0.74), geboren (0.67)
geboren	hij (0.74), zij (0.67)
is	zij (1.0), persoon (0.88), hij (0.74), geboren (0.67), huishouden (0.67)

Figure 3.16: Cooccurrence probabilities $P(u|s)$ above 0.5 in the source sentence, with u being an untranslatable word, and s any support word from the sentence.

Here is how the algorithm builds up the translation frame: Firstly, all untranslatable words are included in the translation frame.

(3.17) "kunt (...) welk (..) hij (...) zij geboren is → Please (...) birth"

Secondly, it adds all support words with a probability above 0.5.

(3.18) "kunt (...) persoon (...) huishouden aangeven (...) hij (...) zij geboren (...) → please indicate (...) person (...) household"

Thirdly, the translations of these support words are also included to the frame. However, the English words "indicate", "person" and "household" are already in the frame. The same holds for the Dutch words "aangeven", "persoon", and "huishouden". The support words "hij", "zij" and "geboren" on the Dutch side, and "birth" on the English side are untranslatable. Thus, the translation frame remains unchanged by this step.

Other words must be included in the frame, because their alignment cannot be disambiguated by our heuristics. Thus, the Dutch word "van" and the two occurrences of the word "of" are included in the sentence frame. The same happens to Dutch "de" and the two occurrences of "the". Inversely, there are two translations for the English

word "your", namely Dutch "u" and "uw". Since our dictionary cannot distinguish between the correct alignment ("your"–"uw") and the incorrect one, all these words are included in the frame. Finally, the Dutch word "of" (English: "or") is misinterpreted as a translation of either occurrence of the English "of". The resulting translation frame is:

(3.19) "kunt u van de X_2 persoon in uw huishouden aangeven in welk X_6 hij of zij geboren is → please indicate the X_6 of birth of the X_2 person in your household"

Such a translation frame can yield perfect translations for sentences of the following kind: "Please indicate the date/month/year of birth of the third/fourth/fifth/... person in your household." (sentences are taken from the corpus).

The intra-sentential alignments of the word "please" to "birth", "person" and "household" are an indication that our corpus is insufficiently large for computing representative distributions. Nevertheless, the example shows how our translation frame generation algorithm can deal with real data, if it is not hampered by inconsistencies between word and phrase lookup.

4 Summary and Conclusions

In this thesis, we have examined the problem of machine translation from a theoretical as well as a practical angle. In Chapter 1, we have presented an overview of the research field, in Chapter 2 we have built a basic translation system capable of two-phase as well as "pure" EBMT, and in Chapter 3 we have developed a new way of generating translation frames on the basis of a two-dimensional alignment model.

The field of machine translation is huge – but the translation quality is still not sufficient to yield the expected practical benefit. What used to be two distinct translation paradigms – SMT and EBMT – has long been mingled into a variety of different methods. Recent systems can no longer be classified as either SMT or EBMT. If most authors still claim their system is either EBMT or SMT, that merely expresses in which line of research tradition they place their work. Judging from their methodology, most systems after (Brown et al., 1990) and (Nagao, 1984) are in some way a synthesis of the two approaches.[1]

Placing our work in the EBMT tradition, we have built a translation system capable of "pure" as well as two-phase EBMT. The lexicon of our translation system has been obtained using statistical processing as well as linguistic resources. More specifically, the *phrase table* contains 1-on-1 aligned phrases, which have been extracted from the corpus on the basis of statistical alignment, while the *dictionary* contains single-word translations from two freely available online dictionaries. For "pure" EBMT, we implemented a *runtime* translation method, which is exclusively based upon the comparison between input sentence and translation example. For two-phase EBMT, we implemented a *compiled* translation method, which relies on pre-processed translation examples (templates) as input. We further implemented two extensions of the runtime translation method, namely a *phrase enlargement* and a *matched sentence cut* algorithm. In order to be able to compare both translation methods, we have generated translation rules for the compiled approach by replacing all occurrences of phrase table entries by variables.

Since we built the translation system from scratch and on the basis of a very small corpus, we could not expect a good translation result. Accordingly, the overall experimental results of both approaches were disappointing with only 15% correct translations retrieved by either approach. However, our aim had not been to build a state-of-the-art translation system, but to gain a better understanding of the different translation methods. As to the runtime translation method, our experiments have shown the effectiveness of both of our extensions (phrase enlargement and matched sentence cut). As to the

[1] (Lepage & Denoual, 2005a) is an exception.

compiled translation method, our experiments revealed good and bad kinds of template. In particular, some of our templates were *counter-productive templates* as they consisted of an empty source side and a non-empty target side (e.g. "$X_1X_2 \rightarrow X1$ about X_2"). Other templates were not beneficial for template matching, because they were no longer *representative* for the original translation example. The best translation performance was achieved by the compiled translation method with translation rules and lexicon perfectly complementary to each other. Under these conditions, our implementation of two-phase EBMT largely outperformed our "pure" EBMT approach.

Based on these insights we re-approached the problem of template generation from a theoretical perspective. Firstly, the lexicon should have a crucial influence on translation rule generation, as well as the source and target language. Secondly, we explicitly aimed at a *sensitive* preprocessing method that would maintain the template's representativity for the translation example. Such representative translation templates, which incorporate the structural discrepancies between the source and target sentence, we call *translation frames*. In contrast to previous approaches, we propose to build them on the basis of a *two-dimensional alignment model*. Such an alignment model enables the alignment of *untranslatable* words that would normally be aligned to the empty cept to be *aligned indirectly* via intra-sentential dependencies. Our *prototype system* establishes intra-sentential alignments statistically and inter-sentential alignments via the lexicon. Our translation frame algorithm includes all those words in the frame that are involved in an indirect alignment.

We believe that this way of template generation has a high potential for the following practical advantages: Firstly, the method enables the usage of single-word translations for translation frame generation. This is of great advantage given that bilingual dictionaries are widely available and word translations can be extracted from bilingual corpora more effectively than phrase translations. At the same time, our method does not require a translation for every word in the sentence, allowing for the fact that some words are *untranslatable*. Secondly, our method allows for intra- and inter-sentential dependencies to be computed independently of one another, and with independent methods. For example, intra-sentential dependencies can be computed on monolingual corpora, while inter-sentential dependencies can employ bilingual dictionaries. Thus, on the whole our method offers a great flexibility.

Unfortunately, the potential of our method has so far not materialised in practical experiments. Again, a "good" overall translation performance was not to be expected due to limited resources. For example, our experiments revealed that the amount of data from which we have calculated the probabilities for the intra-sentential dependencies was insufficient. As to comparing the translation performance to that of the baseline system from Chapter 2, it must be borne in mind that our translation frame generation algorithm uses the lexicon in quite a different way. First of all, the baseline system's performance depends on phrase translations, while the quality of the translation frames depends on the quality of single-word translations. This makes it impossible to run experiments comparing the performance of the two translation rules, which do not at the

same time also compare the quality of phrase and single-word translations in our lexicon. Furthermore, the combined use of single-word translations (during pre-processing) and phrase translations (during translation) demands the lexicon to provide consistent phrase and single-word translations for non-crossing 1-on-1 aligned phrases. Since the dictionary component of our lexicon does not support phrase lookup, this requirement is not always fulfilled. The baseline methods are hardly affected by this shortcoming of our lexicon, but they are rarely able to make use of the dictionary in the first place. Finally, the translation frame algorithm itself is affected by lexicon entries in two ways: On the one side, words that are generally translatable but do not have an entry in the lexicon will automatically remain in the translation frame – even together with their support words. That way, the translation frame may become virtually unmatchable because it is too specific. On the other side, imperfect translations, as we suspect they are contained in the phrase table, may also hamper performance. They prevent indirect alignment of untranslable words. Thus, our method requires a lexicon that provides high quality translations of translatable words, not one that provides low quality translations for all words. The distinction between translatable and untranslatable words should be examined further in future work.

We have developed an EBMT approach which strongly relies on an alignment model, which is more typical for SMT systems. Moreover, this alignment model can be computed completely statistically as in (Fraser & Marcu, 2007), or – as in our prototype system – contain statistical elements which enable a probabilistic distinction between relevant and irrelevant parts of the example translation. However, in contrast to SMT, we use the alignment model exclusively for the analysis of existing translation examples. Since we do not use the alignment model to generate translations, we can afford to align words without any restrictions on word order and still do not run into computability problems.[2] The liberal language model we use to model intra-sentential dependencies does not calculate the probabilty of bags of words, but that of the co-occurrence of two words regardless of their positions within the sentence. We regard this as a great advantage, because distortion models as they have been developed in SMT are exclusively based on word position, which is not a reliable indicator: For example, three words can be part of a phrase as well as they can make up a complete clause. In our approach, the generation of new translations is based on translation frames, which are pre-processed translation examples. This makes it possible to store structural discrepancies locally in translation examples, thereby avoiding the need for a global model. Further, the use of translation frames neutralises any alignment to the empty cept. If a word cannot be aligned directly, it will be aligned indirectly; if it cannot be aligned indirectly, this word will always end up being aligned to the overall translation frame. We believe our method capitalises on the strength of both research traditions in this way.

However, so far, the word order problem is not fully solved by translation frames.

[2] Also consider that our approach only requires to calculate the intra-sentential dependencies for untranslatable words – not for all words.

This is because, in this thesis, we have focused on untranslatability as an indicator for structural discrepancies. In contrast, word order divergences are not captured explicitly. Since many word order divergence phenomena are also linked to untranslatability, one might hope that there is enough correlation between the two factors such that word order divergences are largely covered by our solution to untranslatability. However, there are a certain number of sentence pairs without untranslatable words, which are still not 1-on-1 translations. An example is the German-Dutch translation of the subclause:

(4.1) ", dass ich die Katze gesehen habe. →, dat ik de kat heb gezien."

The determination of the crossings in this alignment is a non-trivial task, which we leave to future work.[3]

In this thesis, we have presented an approach on the basis of *untyped* templates. The expressive power of this model is limited by the fact that a balance must be maintained between the proportion of words of the sentences that are included in the translation frame and the proportion of words replaced by variables. If the frame contains too many words, it will be too specific to match on an unseen input sentences. If the words are too few, it will no longer be representative and match on too many input sentences. The use of typed templates would introduce a more sensitive form of generalisation: to remove the word but still keep the type. While that balance between "representative" and "general" must be maintained in any case, an "upgrade" to typed templates would loosen the restriction considerably.

[3]The general approach we would suggest here, is:

1. Determine the closest non-crossing one-on-one alignment.
2. Compare the alignment from step 1 to the actual word order in the example.

Bibliography

Ahmed, A., & Hanneman, G. (2006). *Syntax-based statistical machine translation: A review.* (Unpublished manuscript, School of Computer Science, Carnegie Mellon University, http://www.cs.cmu.edu/~amahmed/papers/Ahmed-Hanneman-survey-SSMT.pdf)

Bod, R. (2007, September). Unsupervised syntax-based machine translation: The contribution of discontiguous phrases. In *Proceedings of MT summit XI.* Copenhagen, Denmark.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., et al. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79–85.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Mercer, R. L., et al. (1988, June). A statistical approach to French/English translation. In *Proceedings of the second international conference on theoretical and methodological issues in machine translation of natural languages.* Carnegie Mellon University, Pittsburgh.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, *19*(2), 263–311.

Brown, P. F., Pietra, J. D., Jelinek, F., Mercer, R. L., & Roossin, P. S. (1988). A statistical approach to language translation. In *Proceedings of the twelfth international conference on computational linguistics* (pp. 71 –76). Budapest, Hungary.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., Lafferty, J. D., & Mercer, R. L. (1992). Analysis, statistical transfer, and synthesis in machine translation. In *Proceedings of the fourth international conference on theoretical and methodological issues in machine translation of natural languages* (pp. 83 –100). Montreal, Canada.

Carl, M. (1998). A constructivist approach to machine translation. In *Workshop on new methods in language processing and computational natural language learning.*

Carl, M. (1999). Inducing translation templates for example-based machine translation. In *Proceedings of MT summit VII* (pp. 250–258). Singapore.

Carl, M. (2001, September). Inducing translation grammars from bracketed alignments. In *Proceedings of the workshop on example-based machine translation, MT summit VIII* (pp. 12–23). Santiago de Compostela, Spain.

Carl, M. (2006). A system-theoretical view of EBMT. *Machine Translation*, *19*(3–4), 229–249.

Carl, M., & Hansen, S. (1999, September). Linking translation memories with example-

based machine translation. In *Proceedings of MT summit VII* (pp. 617–624.). Singapore.

Carl, M., & Way, A. (2003). Introduction. In *Recent advances in example-based machine translation.* Springer.

Charniak, E., Knight, K., & Yamada, K. (2003, September). Syntax-based language models for statistical machine translation. In *Proceedings of MT summit IX.* New Orleans, USA.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings ACL 2005.*

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, *33*(2), 201–228.

Cicekli, I. (2005, September). Learning translation templates with type constraints. In *Proceedings of example-based machine translation workshop, MT summit X* (pp. 27–34). Phuket, Thailand.

Cicekli, I., & Güvenir, H. A. (2001, July–August). Learning translation templates from bilingual translation examples. *Applied Intelligence*, *15*(1), 57 – 76.

Collins, B. (1998). *Example-based machine translation: An adaptation-guided retrieval approach.* Unpublished doctoral dissertation, University of Dublin, Trinity College, Department of Computer Science.

Collins, B., & Cunningham, P. (1996). Adaptation guided retrieval in EBMT: A case-based approach to machine translation. In *Proceedings of the third european workshop on advances in case-based reasoning* (pp. 91–104). Springer-Verlag.

Collins, B., & Cunningham, P. (1997). Adaptation-guided retrieval: Approaching EBMT with caution. In *Proceedings of the 7th international conference on theoretical and methodological issues in machine translation* (pp. 119–126). Santa Fe, New Mexico, USA.

Cowan, B., Kucerova, I., & Collins, M. (2006). A discriminative model for tree-to-tree translation. In *Proceedings of EMNLP 2006.* (pp. 232–241). Association of Computational Linguistics.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc.*.

Denoual, E. (2006). *Méthodes en caractères pour le traitement automatique des langues.* Docteur spécialité informatique, Université Joseph Fourier, Grenoble, France. (chapter III/2)

Dorr, B. J. (1994). Machine translation divergences: a formal description and proposed solution. *Computational Linguistics*, *20*(4), 597–633.

Dyvik, H. (2005). Translations as a semantic knowledge source. In *Proceedings of the second baltic conference on human language technologies.* Tallinn, Estonia.

Fox, H. (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of conference on empirical methods in natural languag processing (EMNLP).*

Fraser, A., & Marcu, D. (2007, June). Getting the structure right for word alignment: LEAF. In *Proceedings of the 2007 joint conference on empirical methods in nat-*

ural language processing and computational natural language learning (EMNLP-CoNLL) (pp. 51–60). Prague, Czech Republic.

Groves, D., & Way, A. (2005). Hybrid example-based smt: The best of both worlds? In *Proceedings of the ACL-05 workshop on building and using parallel texts: Data-driven machine translation and beyond* (pp. 183–190). Ann Arbor, USA.

Huang, B., & Knight, K. (2006). Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics* (pp. 240–247). Morristown, NJ, USA: Association for Computational Linguistics.

Hutchins, J. (2005, September). Towards a definition of example-based machine translation. In *Proceedings of workshop on example-based machine translation, MT summit X* (pp. 63–70). Phuket, Thailand.

Kaji, H., Kida, Y., & Morimoto, Y. (1992). Learning translation templates from bilingual text. In *Proceedings of the 14th conference on computational linguistics* (Vol. 2, pp. 672–678). Nantes, France: Association for Computational Linguistics.

Koehn, P. (2004, August). A beam search decoder for phrase-based statistical machine translation models [Computer software manual]. (User manual and description for version 1.2.)

Koehn, P., Och, F. J., & Marcu, D. (2003, May). Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003.* Edmonton, Canada.

Lepage, Y. (1998). Solving analogies on words: an algorithm. In *Proceedings of the 17th international conference on computational linguistics* (pp. 728–734). Morristown, NJ, USA: Association for Computational Linguistics.

Lepage, Y. (2005). Translation of sentences by analogy principle. *Archives of Control Science*, *15*(4), 585–594.

Lepage, Y., & Denoual, E. (2005a, September). The 'purest' ever built EBMT system: No variable, no template, no training, examples, just examples, only examples. In *Proceedings of workshop example-based machine translation, MT summit X* (pp. 81–90).

Lepage, Y., & Denoual, E. (2005b, December). Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, *19*(3–4), 251–282.

Lopez, A., & Resnik, P. (2006). Word-based alignment, phrase-based translation: what's the link? In *Proceedings of the 7th conference of the association for machine translation in the americas: visions for the future of machine translation* (pp. 90–99). Boston, USA.

Mansell, R. (2005). Optimality in translation. In *New research in translation and interpreting studies.* Tarragona (Spain).

Marcu, D. (2001). Towards a unified approach to memory- and statistical-based machine translation. In *Proceedings of the 39th annual meeting on association for computational linguistics* (pp. 386–393). Toulouse, France.

Marcu, D., & Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP* (pp. 133–139).

McTait, K. (2001, September). Linguistic knowledge and complexity in an ebmt system based on translation patterns. In *Proceedings of the workshop on ebmt, MT summit VIII.* Santiago de Compostela, Spain.

Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn & R. Banerji (Eds.), *Artificial and human intelligence* (chap. 11). NATO.

Nitta, Y. (1986). The idiosyncratic gap: A tough problem to structure-bound machine translation. In *Proceedings of the 11th international conference on computational linguistics* (pp. 107–111).

Och, F. J., & Ney, H. (2000, October). Improved statistical alignment models. In *ACL00* (pp. 440–447). Hongkong, China.

Och, F. J., & Ney, H. (2004, December). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30*(4), 417–449.

Prince, A., & Smolensky, P. (2002). *Optimality theory: Constraint interaction in generative grammar.* Blackwell Publishing. (ROA Version, 8/2002)

Quirk, C., & Menezes, A. (2006). Do we need phrases?: challenging the conventional wisdom in statistical machine translation. In *Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics* (pp. 9–16). Morristown, NJ, USA: Association for Computational Linguistics.

Sato, S., & Nagao, M. (1990). Toward memory-based translation. In *Proceedings of the 13th conference on computational linguistics* (pp. 247–252). Morristown, NJ, USA: Association for Computational Linguistics.

Schuch, A. (2008, July). *The usage of the particle ba in mandarin chinese: A computational corpus analysis.* Available from `http://student.science.uva.nl/~aschuch/ChineseBA_ASchuch.pdf`

Simard, M., Cancedda, N., Cavestro, B., Dymetman, M., Gaussier, E., Goutte, C., et al. (2005, October). Translating with non-contiguous phrases. In *Proceedings of the human language technology conference conference on empirical methods in natural language processing (HLT/EMNLP 2005).* Vancouver, B.C., Canada.

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

Somers, H. (2001, September). EBMT seen as case-based reasoning. In *Workshop on example-based machine translation, MT summit VIII.* Santiago de Compostela, Spain.

Sumita, E., & Iida, H. (1991). Experiments and prospects of example-based machine translation. In *Proceedings of the 29th annual meeting on association for computational linguistics* (pp. 185–192). Morristown, NJ, USA.

Tinsley, J., Hearne, M., & Way, A. (2007). Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. In *Proceedings of treebanks and*

linguistic theories (TLT '07). Bergen, Norway.

Turcato, D., & Popowich, F. (2001, September). What is example-based machine translation? In *Proceedings of the workshop on example-based machine translation, MT summit VIII* (pp. 18–22). Santiago de Compostela, Spain.

Watanabe, T., & Sumita, E. (2003, September). Example-based decoding for statistical machine translation. In *Proceedings of MT summit IX.* New Orleans, USA.

Way, A., & Gough, N. (2005, September). Comparing example-based and statistical machine translation. *Natural Language Engineering*, *11*(3), 295 – 309.

Weaver, W. (1949, July). Translation. In W. N. Locke & A. D. Booth (Eds.), *Machine translation of languages: fourteen essays* (pp. 15–23). Technology Press of the Massachusetts Institute of Technology.

Yamada, K., & Knight, K. (2001, July). A syntax-based statistical translation model. In *Proceedings of the 39th annual meeting of the association of computational linguistics (ACL'01)* (pp. 6–11). Toulouse, France.

Ziegler, C. (2008, November). Rosetta statt Babel – Hauptströmungen der maschinellen Übersetzung. *iX Magazin für professionelle Informationstechnik*(11), 42-47.