# A Part-of-Speech-Based Search Algorithm for Translation Memories

**Reinhard Rapp**

University of Mainz, FASK
76711 Germersheim, Germany
rapp@mail.fask.uni-mainz.de

**Abstract**

The retrieval of related sentences in state-of-the-art translation memory systems is based on orthographic similarities. This often leads to poor search results, since orthographically similar sentences are not necessarily semantically related. In this paper we propose a search algorithm that aims to reduce this problem by taking part-of-speech information into account. It requires that the parallel sentences stored in the translation memory are processed using standard tools for word alignment and part-of-speech tagging. The work described is part of an ongoing project in example-based machine translation.

## 1. Introduction[1]

Translation memories are tools that facilitate the translation of repetitive kinds of text. As a text is translated, all pairs of corresponding sentences in the source and the target language are stored in a database. This database is searched when a new sentence is to be translated. If the sentence is found the respective translation is retrieved automatically. Thus, in principle, by using a translation memory each sentence of a source language needs to be translated only once.

However, in practical texts, due to the almost infinite number of possible sentences, it is rare that the same sentence occurs more often than once. For example, from the 40 000 sentences of the American Brown Corpus only 318 occur two or more times (Rapp, 1998). For this reason, modern translation memory systems not only search for identical but also for similar sentences. Of course, in the case of a fuzzy match, the translation retrieved from the database needs to be edited. Productivity is still increased since in many cases the editing will take less time than to translate the source sentence from scratch.

Currently, the search mechanisms of commercially available translation memory systems are based on the comparison of orthographic similarities between sentences. This facilitates implementation and allows the construction of fast search engines. However, since orthographic similarity does not necessarily mean semantic similarity, it often leads to poor search results. For example, the two German sentences “Montage gefallen mir nicht sehr” (I don’t like Mondays) and “Montagehallen sind nicht leer” (Assembly halls are not empty) are orthographically similar, but have totally different meanings. For this reason it would be desirable to use a search mechanism that takes syntactical and / or semantical information into account.

In this paper, we propose a search algorithm that aims at solving the problem of misleading orthographic similarities while at the same time increasing the chances of finding a good match. In information retrieval terms, recall and precision are to be improved at the same time. The basic idea is to exploit syntactical information as provided by part-of-speech taggers. If the part-of-speech-system is fine-grained enough and the accuracy of the tagger high, then the search mechanism that has so far been applied to words can be applied to parts of speech. Since the number of different parts of speech is lower than the number of different words, the chances of finding a good match are better.

The paper is organized as follows: First, we look at search methods used in translation memories and in systems for example-based machine translation (EBMT) as described in the literature. We then give an overview of the EBMT-project currently running at our University. Finally, we describe the search algorithm proposed in this framework in detail.

## 2. Current search algorithms

There are mainly four techniques used for the retrieval of similar sentences in translation memory and EBMT-systems:

1. fuzzy matching
2. syntax trees
3. thesauri
4. neural networks

### 2.1. Fuzzy matching

This is the dominating approach in leading commercial translation memory systems like Trados’ Translators’ Workbench, Star’s Transit, IBM’s Translation Manager or Atril’s Déjà Vu (only ZERES*TRANS* uses linguistics). The method is based on orthographic similarities, which can be efficiently computed by comparing the number of corresponding substrings (e.g. bi- or trigrams) of two sentences (Angell et al., 1983; Heitland, 1994; Rapp, 1997). Fig. 1 shows an example where the number of bigrams common to two strings is used as a measure for their similarity.

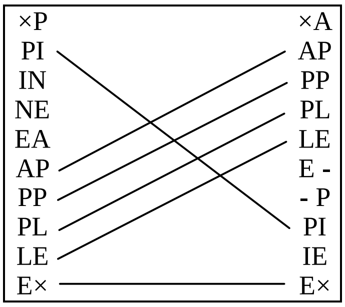


Figure 1. Bigrams common to the two strings *pineapple* and *apple-pie*. Since six out of ten bigrams correspond, the similarity is 60%. (The symbol ‘×’ is added to give all characters the same weight.)

[1] This research was supported by the DFG.

## 2.2. Syntax trees

This approach requires parsers for both languages to be considered. The parse tree of the sentence to be translated is compared to the parse trees of all source language sentences in the translation memory. If an identical parse tree is found, it is assumed that the parse tree of the correct translation should be identical to the parse tree of the corresponding target language sentence retrieved from the translation memory (Maruyama & Watanabe, 1992). The main problem with this approach is that high quality parsers for unrestricted language are not available for many languages. Also, the disambiguation of semantically ambiguous words is not always possible by considering syntax only.

## 2.3. Thesauri

The problem of ambiguity is better accounted for in the thesaurus-method (Sata & Nagao, 1990; Sumita et al., 1990). By use of a thesaurus it is determined in how far one word of a language can replace another one in a sentence without changing the meaning of the sentence. In addition, sentence structure is taken into account by considering dependency trees. Thus, two sentences or sentence fragments can be compared by comparing the similarity of the words at corresponding positions of their dependency trees. The construction of a suitable thesaurus can be facilitated or possibly replaced by corpus-based automatic methods for computing word similarities as suggested by Grefenstette (1994), Ruge (1995), Schütze (1997), Lin (1998) and others.

## 2.4. Neural networks

McLean (1992) suggested the use of a neural network for EBMT (see fig. 2). The network is supposed to learn the relations between the sentences of a source language and their corresponding translations in the target language. Hereby, the source language sentences are applied to the input layer of the neuronal network in such a way that each word relates to one of 30 neurons. Each of the 30 neurons corresponds to one word of a vocabulary, i.e. the vocabulary is restricted to 30 words and a localistic representation is used in the input layer. The representation in the output layer is also localistic, but in this case not a word but a target language sentence corresponds to each neuron.

The network is trained by applying a source language sentence to the input layer and the corresponding target language translation to the output layer of the network and by adjusting the weights between the two layers using the delta-rule (Rumelhart & McClelland, 1986). When this process of supervised learning is repeated with many sentence pairs, it is hoped that the network will be able to generalize correctly. This means that if during recall a sentence different from any of the trained sentences is applied to the input layer, in the output layer the neuron corresponding to the best fitting translation should be activated.

Although the model is reported to work in principle, there are a number of serious problems with it. First, the number of neurons in the input layer increases quadratically with sentence length and vocabulary size. Secondly, the system puts too much emphasis on the absolute positioning of words in the source sentence, i.e. insignificant omitions or insertions of words tend to lead to overreactions. Third, since the relations between source- and target-sentences are coded only internally, the system does not give any indication as to what changes should be made to a target sentence when there is a discrepancy between the source sentence to be translated and the most similar trained sentence.

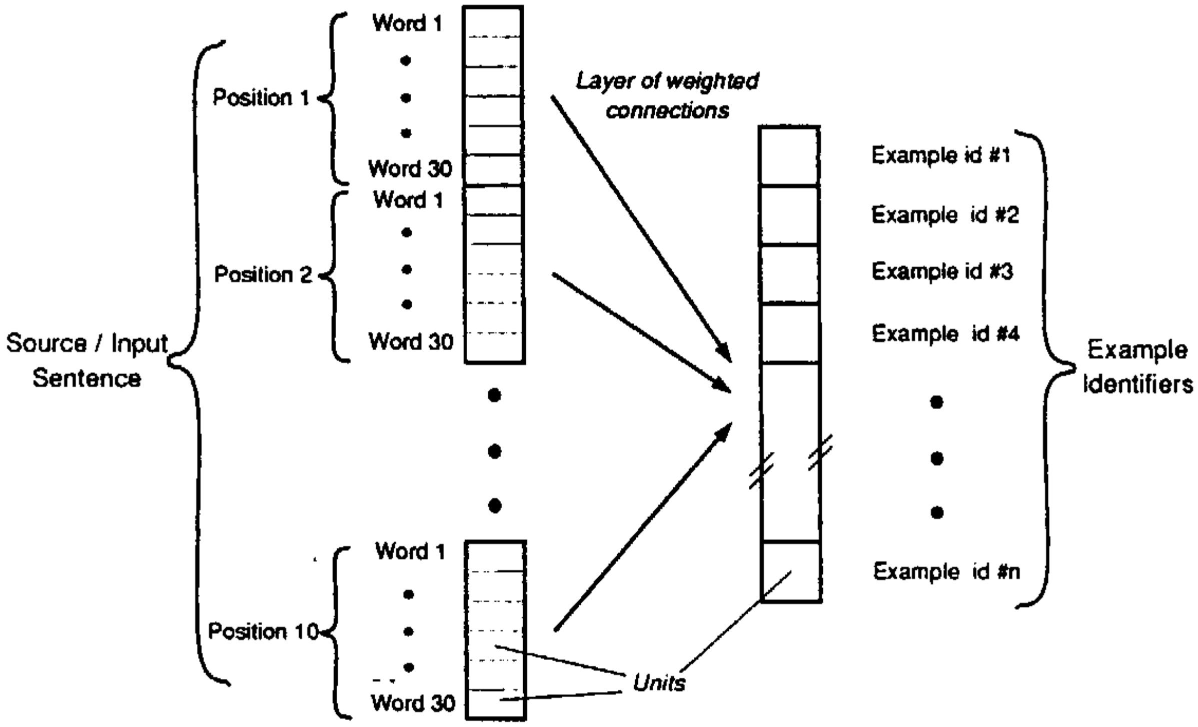


Figure 2. EBMT-system based on neural networks as proposed by McLean (1992).

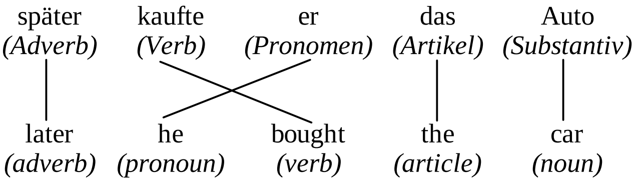
## 3. EBMT framework

The search algorithm described in this paper is part of a larger EBMT project which aims at using established corpus-statistical algorithms for machine translation. Figure 3 gives an overview of the project. Programme modules are characterized by rectangles, linguistic data by ovals. Directed connections signify the order of processing.

The units belonging to the core of the translation system are shaded light gray (left and lower part of figure 3). The linguistic resources required for translation are marked dark gray, namely a database of word-aligned and POS-tagged parallel sentences, a bilingual dictionary, two POS-dictionaries, and collections of collocations and frequent co-occurrences. The white units show how these resources can be generated automatically from raw corpus data.

## 4. A POS-based search algorithm

The proposed search algorithm is based on the assumption that the sentence pairs in the translation memory have been correctly word-aligned and tagged, as shown in the following example:



Please note the different word order in German and English for this type of sentence starting with an adverb. Although there is almost no orthographic similarity, a source language sentence like “dann bereitete er das Essen” (then he prepared the meal) would match with this example, since what we look at is the sequence of parts of speech which is identical. From the word order information in the translation memory it would be correctly concluded that in the translation of this sentence a transposition between the verb and the pronoun is required.

Of course, in practice there is no guarantee that our search for a particular part-of-speech sequence in the translation memory will be successful. Before we discuss the complications arising from this, let us first briefly consider the steps necessary to automatically generate a translation memory in the required form (i.e. with word alignments and part-of-speech tags). Assuming that we start with a parallel corpus, three steps are necessary:

- Sentence alignment
- Word alignment
- Part-of-speech tagging

### 4.1. Sentence alignment

Sentence alignment means to explicitly determine the pairwise correspondences of sentences or groups of sentences in a parallel corpus. Most algorithms described in the literature start by generating a large number of possible alignments and then select the best one by applying an evaluation function to each alignment. Established evaluation functions include:

1. *Sentence length:* Those alignment is considered optimal where the average length difference of corresponding source/target sentences is minimal.
2. *Orthographic similarity:* The optimal alignment is the one that maximizes the orthographic similarity between corresponding sentences.
3. *Dictionary lookup:* If a dictionary is available, it can be looked up how many of the words in a target language sentence are listed as translations of words in the corresponding source language sentence. The alignment that maximizes the number of matches is considered optimal.

Surprisingly, the first method – although hardly using any language specific information – has been reported to give accuracies of around 99% for the parallel Hansard corpus, i.e. the proceedings of the Canadian parliament. The second method is applicable to closely related languages only, while the third method should be the most accurate and robust but requires a dictionary.

### 4.2. Word alignment

Given a sentence-aligned parallel corpus, word alignment can be considered as a combinatorial problem. For example, from the sentence pair “*Hans arbeitet / Jack works*” it can be concluded that the translation of *Hans* is either *Jack* or *works*. If a further sentence pair “*Hans schläft / Jack sleeps*” is available, the translation *works* for *Hans* can be ruled out. Thus, for each word a single possibility remains, i.e. *Hans* is to be translated as *Jack*, *arbeitet* as *works*, and *schläft* as *sleeps*.

However, a pure combinatorial approach is not easily put into practice because sentences tend to be long, translations free, and words ambiguous. More tolerant is a statistical approach. If the translation of a word is to be determined statistically, all source language sentences containing this word are considered. It can be expected that the chances of observing the correct translation in the corresponding target language sentences are much higher than expected from chance and especially when compared to the remaining target language sentences. Comparing the observed frequencies in corresponding versus non-corresponding target language sentences or testing for significance will help to quantify this.

These results give us some measure for the probability that a certain word of the target language is the translation of a particular word of the source language. This information is very useful for word alignment: Of all possible word alignments for a given pair of sentences we simply select the one that maximizes the probabilities that the aligned words are translations of each other. However, for infrequent words, where probability estimates are poor, the results may be unsatisfactory. We therefore propose to use a mix of the statistical and the combinatorial approach. How this can be successfully implemented using a spreading activation type of algorithm was shown in a previous publication (Rapp, 1996:108).

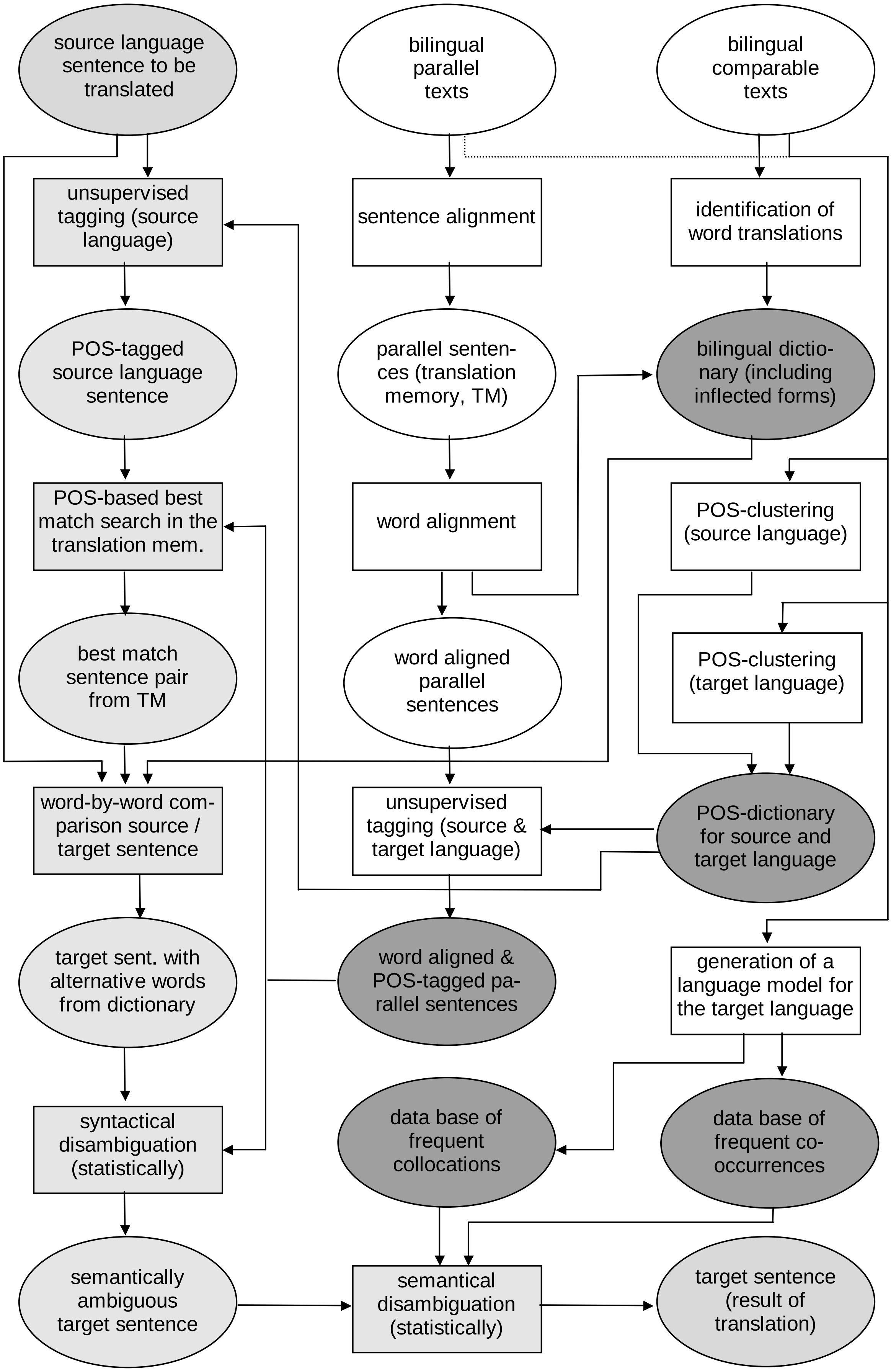

source language sentence to be translated
bilingual parallel texts
bilingual comparable texts
unsupervised tagging (source language)
sentence alignment
identification of word translations
POS-tagged source language sentence
parallel sentences (translation memory, TM)
bilingual dictionary (including inflected forms)
POS-based best match search in the translation mem.
word alignment
POS-clustering (source language)
best match sentence pair from TM
word aligned parallel sentences
POS-clustering (target language)
word-by-word comparison source / target sentence
unsupervised tagging (source & target language)
POS-dictionary for source and target language
target sent. with alternative words from dictionary
word aligned & POS-tagged parallel sentences
generation of a language model for the target language
syntactical disambiguation (statistically)
data base of frequent collocations
data base of frequent co-occurrences
semantically ambiguous target sentence
semantical disambiguation (statistically)
target sentence (result of translation)

Figure 3. Architecture of the EBMT-system.

### 4.3. POS-tagging

Although successful implementations of rule-based (Samuelson & Voutilainen, 1997) or neural-network-based (Schmid, 1994) types of taggers exist, we want to concentrate here on the popular statistical taggers, since they probably offer the best compromise between development effort and tagging accuracy.

Statistical taggers need information on the transition probabilities between tags, which can – for example – be derived from a manually tagged corpus. When tagging a new text it is assumed that ambiguities should be resolved in such a way that the tag transition probabilities are maximised. This approach is rather successful, in particular if the so called lexical probabilities are also taken into account, i.e. the probability that a certain word form assumes a particular tag (without considering context).

Since large enough tagged corpora to derive the probabilities from are not always available, algorithms for unsupervised tagging have been suggested (Cutting et al., 1992; Merialdo, 1994). Simply speaking, they iteratively improve the tagging of a corpus by changing the tags of ambiguous words in such a way that observed patterns of tag sequences are emphasized. The bootstrapping works because many words are unambiguous and chances of correct guesses for the others are good.

### 4.4. Possible search results

As mentioned above, our POS-based search in the translation memory may not always lead to the desired result. Possible outcomes are:

- Exactly one matching sentence is found
- Several matching sentences are found
- No matching sentence is found

If we have exactly one match, the retrieved translation can directly serve as the pattern for the new translation. Likewise with several matches, except that we now have the choice between several patterns that may be appropriate. The selection can – for example – be based on the number of identical words, on the degree of orthographic similarity (see 2.1.) or on word similarities derived from a thesaurus or a vector space model (see 2.3.).

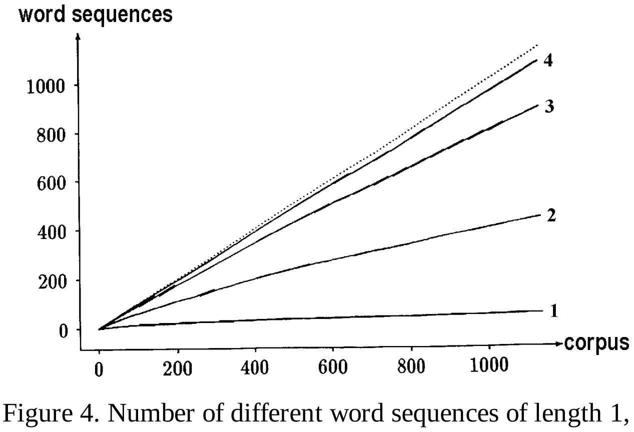


Figure 4. Number of different word sequences of length 1, 2, 3, and 4 in the Brown corpus depending on corpus size (all coordinates × 1000). The almost linear curves indicate a very large number of possible word sequences.

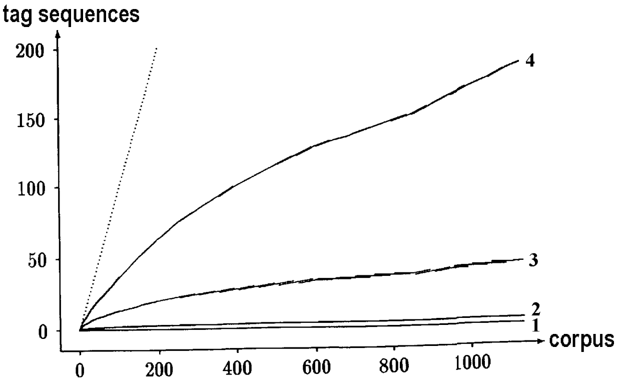


Figure 5. Number of different tag sequences of length 1, 2, 3, and 4 in the Brown corpus depending on corpus size (all coordinates × 1000). The asymptotical curves indicate a limited number of possible tag sequences.

If no full match is found at all, the search can be extended to matches of partial sentences, where punctuation or conjunctions can serve as delimiters. Of course, the use of several partial translation patterns involves the risk that the parts do not fit together properly, i.e. that the resulting translation will be ungrammatical. However, the fact that our search mechanism is based on parts of speech instead of words greatly improves the chances of getting a full match. This is confirmed by figures 4 and 5, which present statistics derived from the American Brown corpus on the observed number of different word- or tag-sequences depending on corpus size. (Please note the different vertical scales of both figures.)

## 5. Translation and disambiguation

In the example from the beginning of the previous section we wanted to translate the sentence "dann bereitete er das Essen" (then he prepared the meal) into English. Our search in the translation memory gave us an appropriate tag sequence for the target sentence together with word order information.

### 5.1. Dictionary lookup and generation

What is further needed is a dictionary based on word forms that also includes part of speech-information. Let us assume by looking up the words of the source sentence we obtained the data from the dictionary as shown in table 1. Please note that the verb *bereitete* is twofold and the noun or proper noun *Essen* is threefold ambiguous. By taking into account the information on word alignment from the translation memory we can construct six possible translations:

1. then he prepared the meal
   (*adverb - pronoun - verb- article - noun*)
2. then he caused the meal
   (*adverb - pronoun - verb - article - noun*)
3. then he prepared the food
   (*adverb - pronoun - verb - article - noun*)
4. then he caused the food
   (*adverb - pronoun - verb - article - noun*)
5. then he prepared the Essen
   (*adverb - pronoun - verb - article - proper noun*)
6. then he caused the Essen
   (*adverb - pronoun - verb - article - proper noun*)

| GERMAN | ENGLISH |
|---|---|
| dann | then (adverb) |
| bereitete | prepared (verb)<br>caused (verb) |
| er | he (pronoun) |
| das | the (article) |
| Essen | meal (noun)<br>food (noun)<br>Essen (proper noun) |

Table 1. Results of dictionary lookup.

### 5.2. Syntactical disambiguation

From our translation memory-lookup we know that the tag-sequence of the correct translation must be "*adverb* - *pronoun* - *verb* - *article* - *noun*". This information allows us to rule out translations 5 and 6, which interpret *Essen* as the name of a city. Thus four possibilities remain.

### 5.3. Semantical disambiguation

The remaining ambiguities can only be resolved on semantical grounds. Although this is notoriously difficult, a matrix of word co-occurrences derived from a text corpus allows to make decisions which are at least better than chance (see table 2). For the words relevant to our example, we find the highest co-occurrence frequency between *meal* and *prepared* (48 co-occurrences). Next follow *food* and *prepared* (12), *food* and *caused* (3), and finally *meal* and *caused* (2). This means that the translations *meal* and *prepared* would be selected in this case, which leads to the correct result.

|  | Essen | food | meal | prepared |
|---|---|---|---|---|
| caused | 2 | 3 | 2 | 1 |
| Essen |  | 1 | 1 | 0 |
| food |  |  | 5 | 12 |
| meal |  |  |  | 48 |

Table 2. Matrix of co-occurrence frequencies of words.

## 6. Conclusions

The paradox situation in professional translation today is that the complex systems for fully automatic machine translation are of little use, whereas simple translation memory tools are successfully used by almost everybody working in the technical domain. What we have described here is in essence the outline of a hybrid system that tries to pick the best from both worlds. This is ongoing work, and the current status does not allow us to predict in how far it will be possible to actually achieve the goals. A serious problem is that only a few smaller parallel corpora are readily available for our language pair German – English (e.g. the Proceedings of the European Parliament, see Armstrong et al., 1998). We hope, however, that in the long run the advantages of the data-driven approach to machine translation will pre-dominate (Sumita et al., 1990):

1. *Reduced computational effort* compared to rule-based systems, since the application of syntactical, semantical, and transfer rules is replaced by computing similarities.
2. *Less effort for system development:* The construction of a linguistic rule base is difficult and can only be done by experts, whereas the collection of a large database of translation examples is much easier.
3. *Less effort for the improvement of the translation quality:* The effects of changes to rules in a rule-based system are hard to predict, because there can be complicated interactions between rules. There is no such problem with adding examples to a translation memory.
4. *Context sensitive translation:* Each example in the translation memory can be supplemented with context information, i.e. concerning the field, the speaker or the situation, which can be taken into account when retrieving an example. With rule-based systems this is not so straight forward.
5. *Robustness:* Rule-based systems require an exact match with the rule base. If such a match is not found, no sensible results can be expected. Since the data-driven approach is based on similarities, there will always be a second or third choice.

## 7. References

Angell, R.C., Freund, G.E., Willett, P. (1983). Automatic spelling correction using a trigram similarity measure. *Information Processing & Management*, 19(4), 255–261.

Armstrong, S.; Kempen, M.; Petitpierre, D.; Rapp, R.; Thompson, H. (1998). Multilingual Corpora for Cooperation. *Proceedings of the 1st International Conference on Linguistic Resources and Evaluation (LREC), Granada,* Vol. 2, 975–980.

Cutting, D.; Kupiec, J.; Pedersen, J.; Sibun, P. (1992). A practical part-of-speech tagger. *Proceedings of the Third Conference on Applied Language Processing*, Trento, Italy, 133–140.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Dordrecht: Kluwer.

Heitland, M. (1994). *Einsatz der SpaCAM-Technik für ausgewählte Grundaufgaben der Informatik*. Dissertation an der Universität Hildesheim.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In: *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Vol. 2, 768–773.

Maruyama, H.; Watanabe, H. (1992). Tree cover search algorithm for example-based translation. In: *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, 173–184.

McLean, I.J. (1992). Example-based machine translation using connectionist matching. In: *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, 35–43.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2), 155–171.

Rapp, R. (1996). *Die Berechnung von Assoziationen: ein korpuslinguistischer Ansatz.* Hildesheim: Olms.

Rapp, R. (1997). Text-Detektor. Fehlertolerantes Retrieval ganz einfach. *c't, Magazin für Computertechnik*, 4/97, 386–392.

Rapp, R. (1998). Wortartenorientierte Suche in Translation Memories. In: J. Strässler: *Tendenzen europäischer Linguistik. Akten des 31. Linguistischen Kolloquiums*, Bern 1996. 176–181.

Ruge, G. (1995). *Wortbedeutung und Termassoziation.* Hildesheim: Olms.Sato, S.; Nagao, M. (1990). Toward memory-based translation. *Proceedings of COLING 1990*, 247–252.

Rumelhart, D.E., McClelland, J.L. (eds.) (1986). *Parallel Distributed Processing.* (2 Vols.) Cambridge, MA: MIT Press.

Samuelson, C.; Voutilainen, A. (1997). Comparing a Linguistic and a Stochastic Tagger. Proceedings of the ACL 1997, Madrid.

Sato, S.; Nagao, M. (1990). Toward memory-based translation. *Proceedings of COLING 1990*, 247–252.

Schmid, H. (1994). Part-of-speech tagging with neural networks. In: *Proceedings of the International Conference on Computational Linguistics*, Kyoto, 172–176.

Schütze, H. (1997). *Ambiguity Resolution in Language Learning: Computational and Cognitive Models*. Stanford: CSLI Publications.

Sumita, E.; Iida, H.; Kohyama, H. (1990). Translating with examples: A new approach to machine translation. In: *The Third Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages.* Linguistics Research Center, University of Texas at Austin, 203–212.