CHUNKY:
AN EXAMPLE BASED MACHINE TRANSLATION SYSTEM
FOR SPOKEN DIALOGS

Ralf Engel *

DFKI GmbH, Germany

ABSTRACT

This paper presents an experimental example based machine translation system for spoken German-English dialogs. The system consists of a syntactic analysis, a module for automatic alignment of sentence pairs, a translation module and a generation module. The translation module splits the input sentence into parts and finds the best translations for these parts using the sample corpus and a distance function considering context.

1 INTRODUCTION

1.1 Motivation

This paper presents an experimental system to study example based machine translation (EBMT) for spoken German-English dialogs. The basis for EBMT systems is a sample corpus with aligned bilingual sentence pairs as opposed to rule based systems which rely on manually created translation rules. The sample corpus for the presented system was collected in the Verbmobil project [7] and comprises about 10000 sentence pairs. Verbmobil is a speech-to-speech machine translation system restricted to the domains of scheduling and travel planning. Since the system was developed within a speech-to-speech system, the input sentences always come from a speech recognizer and can therefore be faulty.

The focus of this paper lies on presenting a method for finding the best translation for a given sentence using the sample corpus of the system. Other important modules for translation are not treated here, like an anaphor resolution module or a module for sentence overlapping disambiguation.

1.2 Overview

Example based translation consists of three main tasks. First, the input sentence has to be split into parts, since it is not probable that exactly the same sentence is contained in the sample corpus. Second, for each part the most suitable translation has to be found in the sample corpus. Third, a target language sentence has to be generated from these translated parts. This paper presents solutions for all of these aspects.

The paper is structured in two main parts:

- Preprocessing (offline): including the syntactic analysis (used for preprocessing as well as for translation) and the alignment of the bilingual sentence pairs.
- Translation (online): including splitting of the input sentence into parts, translation of the parts using the sample corpus and generation of the target sentence.

A discussion of the pros and cons of the system concludes the paper.

2 PREPROCESSING

2.1 Syntactic Analysis

Many example based translation systems refrain from using syntactic analysis. But syntactical knowledge is crucial especially for relatively free word order languages like German, since the sequence of words has to be changed in most cases when translating to strict word order languages like English. Also complex sentences containing, e.g., relative clauses, are extremely difficult to handle without any syntactical knowledge. Syntactic analysis is applied to all sentences in the sample corpus (offline) as well as to the input sentence (online).

![Syntactic Tree](image.png)

**Figure 1:** A syntactic tree generated by the parser

---

* This work was funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01IV101K/1. The responsibility for the contents of this study lies with the author.
For the syntactic analysis of both German and English sentences, hierarchical finite state transducers (FSTs) [9] together with a part-of-speech tagger (from IMS, Stuttgart, Germany) are used. FSTs are easy to implement and well-suited for robust parsing [1]. Robustness is important due to the high error rate of the speech recognizer (about 20%). The FSTs generate a simple syntactic structure, consisting mainly of nominal and prepositional phrases and subordinate clauses. Such a rough decomposition of the sentence proved adequate for further processing. Furthermore, the FSTs are non-deterministic, so more than one syntactic structure can result. A sample analysis is shown in Figure 1.

One peculiarity of the analysis should be mentioned here: Each FST can be provided with functions. These functions fulfill tasks which are difficult to model as FSTs, e.g., filling the subcat lists of the verbs. On top of this, they help to speed up the analysis by delaying decisions, e.g., the decision of PP attachment. Like the FSTs, the functions are non-deterministic.

2.2 Preprocessing of the Sample Corpus

The preprocessing of the sample corpus includes the following tasks: Syntactic analysis of all sentences, then splitting the source language sentences into parts, so-called translation units (TUs) and finding the corresponding parts in the target language sentence, so-called target translation units (TTUs). Since the input sentence is syntactically analyzed, each TU/TTU has an internal syntactical structure and TUs and TTUs do not need to consist of contiguous words. An example of two sentence pairs and the assigned TUs and TTUs is shown in Figure 2. Similar concepts to the TUs and TTUs used in this system can be found in [4],[5],[6],[8].

The algorithm developed to find the TUs and TTUs can be looked at as an extended version of the one presented in [3]. With the help of a small lexicon the first connections between the two structures are established. Due to the small size of the lexicon not all words are connected. During the next steps more and more words and phrases are coupled, with the possibility of more than two elements being coupled. The most important steps are:

1. Phrases which include already connected words, are coupled.
2. Unconnected words having comparable part-of-speech (POS) and included in coupled phrases are connected, using a table with corresponding English and German POS.
3. Unconnected phrases fulfilling some heuristics, e.g., same number of words, are coupled, including the words of the phrase.
4. All remaining unconnected words and phrases are coupled into one single TU/TTU-pair.

All words and phrases which are connected together produce an own TU and a corresponding TTU. For an example see Figure 2, where NP₁, NP₅, schlecht, passt are coupled to one TU and NP₀, does, not, suit, NP₇ are coupled to the corresponding TTU. Phrases in the TUs are assigned additional links to the corresponding phrases in the TTUs. This is necessary if more than one phrase is in one TU, since otherwise it is unclear where to insert the translations of the content of the phrases during the construction of the target syntactic tree.

Using this algorithm the TUs tend to be small in size. The advantage is that a wider range of sentences can be translated, since only smaller parts must match between the example sentence and the input sentence. The drawback is that an incorrect usage of the TU/TTU-pair is more likely, because it suffices that a smaller part of both sentences match.

The syntactic analysis of the sample sentences can be ambiguous as mentioned in 2.1. Currently, the effected sentence must be annotated with syntactical markers to force a definite analysis. It is planned to extend the alignment algorithm in order to resolve some of the ambiguities by regarding the different structures between source and target language sentences (see also [3]).

![Table 1: Two sentence pairs from the example corpus. The numbers indicate the different TU/TTU pairs. Some source language phrases have an additional link, e.g., NP₅, connecting to the corresponding target language phrase, e.g., NP₇.](image)

3 TRANSLATION

3.1 Example Based Transfer

The actual translation of an input sentence is now explained in detail. First, the input sentence is syntactically analyzed as described in 2.1. Then, the segmentation of the input sentence into TUs must be handled. The chosen solution is to sort the words in the input sentence according to their frequency of occurrence (less frequent first) and to store them in a list called input wordlist (IWL).
The first word \( w \) is taken from the IWL and a list is created containing all TUs of the sample corpus that fulfill two conditions: the TUs must include \( w \) and all words within the TU must occur in the IWL. The TU, together with its TTU and the corresponding sentence pair, is saved in the list.

Since each of the TUs may have a different TTU which result in different translations, the next task is to choose one TU/TTU-pair for translation, hopefully the one with the adequate translation. To achieve this, the complete example sentence (ES) belonging to each TU is compared with the complete input sentence (IS). Comparing the complete sentences makes sure that all available context within each sentence is considered. The result of the comparison is a distance value and the TU/TTU-pair with the lowest distance value is used for translation.

Distance values must be computed between two words, between two phrases and, finally, between IS and ES.

**Word distance** is the product of the semantic distance between the two words \( w_1, w_2 \) and a weight.

The semantic distance is computed determining the height of the first common node in a semantic word tree, equivalent as described in [2].

The weight is looked up in a four-dimensional table. The dimensions are:

- POS of the current word \( cw \), which has to be translated
- POS of \( w_1 \)
- distance between the \( cw \) and \( w_1 \) in the syntax tree
- within the current TU or not

For space efficiency the table is saved in a compact form. The weights are adjusted manually.

**Phrase distance** is computed by comparing all elements (words and embedded phrases) of one phrase with all elements of the other phrase. The pair with the lowest distance value is chosen first. Then all pairs containing elements of the selected pair are deleted. This procedure is repeated until no pair is left. If no corresponding element can be found for one element, the distance value for this element is looked up in a table which is similar to the word distance weights table.

The result is the sum of the distance values of the chosen pairs and the sum of the values of the elements not included in any pair. The fact that the values are added makes the manual adjustment of the weights and values rather difficult. This is because the number of words in the example sentences may be different, but the computed distance values have to be comparable.

**IS/ES distance** is determined using \( cw \) as a starting point. Then both trees are walked up in parallel until one of the root is reached. The phrase distance of these nodes is returned as the result. Also, a separate weight for each sentence might be useful (not implemented yet), indicating the importance of the correspondence between ES and IS, e.g., more context sensitive translation examples can get a higher weight.

The next step is to delete each word occurring in the TU from the input wordlist (IWL) and repeat the procedure until the IWL is empty. The final result is a list containing TUs and their corresponding TTUs. Additionally, links between the words in the TUs and the corresponding words in the input sentence are saved. These links are important for generation.

**Ordering the words in IWLS**: Back to the already mentioned problem how to order the words in the IWLS. The chosen word order is relevant for the presented algorithms. The reason for this is that the TUs including the same word can differ in size, so different segmentations using differently sized TUs are possible.

The implemented solution prefers less frequent words. This is based on the assumption that such words influence the translation more than more frequent words (like *to be*). Possible alternative solutions are to prefer words with certain POS (e.g., verbs). A more time consuming solution would be to process every word instead of just one and take the one with the highest score.

A completely different approach would be to sort the TUs of the sample corpus instead of the input words and take the first one which matches successfully. But sorting the TUs is a nontrivial problem and it is not guaranteed that it is possible at all.

### 3.2 Generation of the Target Sentence

At this stage we have all the words for the target sentence (from TTUs), but still have to manage the linearisation, i.e., the ordering of the words. First, a syntax tree is build up in the target language, then the words in each phrase are ordered. Finally, the morphology of the words is adapted (this is not described in detail). The flattened syntax tree is then returned as the result of the translation. An example is shown in Figure 3.

**Construction of the syntax tree**: The syntax tree of the source language is taken as a starting point. For each TU/TTU-pair the linked source language words are deleted in the syntax tree and replaced by the target language words. Phrase nodes get a special treatment: If a phrase node in the TU is coupled with one in the TTU, then the original phrase node in the syntax tree is kept. This is necessary, e.g., if an adjective is translated by a separate TU/TTU-pair within a nominal phrase.

Since some words require a greater structural change of the syntax tree after translation, special movement functions can be associated with each TTU offline. So cases like head-switching can be realized. But because of the flat syntactic structure, this is limited to rare cases. A problem is the simultaneous application of functions which can lead to unwanted interaction. This problem is not solved properly yet and left for future work.

Word and phrase movements are not yet automatically detected by the alignment module and a solution to this problem is still under development. The affected words have to be manually handled in the sample corpus and the generated TUs containing these words are ignored. The movement functions are not described in greater detail in this paper.
Ordering the words within a phrase: The way TTUs are inserted into phrases determines the overall word order. This order may be wrong. Therefore, I use FSTs for reordering. The order criterion for most of the words is their POS, but for a few words the word itself is used, e.g., not all adverbs are placed in the same way. The design of the FSTs is not discussed in detail.

4 DISCUSSION

Having presented the system the pros and cons are discussed. An advantage of all example based systems is the simple collection of data. The sentence pairs can be just typed in or taken from an existing corpus. Only little knowledge of the internal processing of the system is needed, as opposed to, e.g., rule based systems. As a consequence, the independence of algorithm and data is higher, i.e., changes to the algorithm do not affect the data. It is furthermore easy to include non-literal translations, especially with the possibility of using large TUs.

In the presented system the processing of languages with relatively free word order and the translation of more complex sentences is better than in other example based systems like in [4] due to the syntactic analysis. Of course writing a grammar is difficult and time consuming.

Some disadvantages of example based systems should not be concealed. It is difficult to handle general complex structural transformations (e.g., converting infinite clauses to subordinate clauses), and since there is no semantic representation available (e.g., in comparison to interlingua systems), semantic disambiguation is not directly supported.

Up to now only a small evaluation with encouraging results has been done for the restricted domain of scheduling and travel planning. But the quality of the example corpus has a strong influence on the results. In the case of the used corpus for the presented system only a manual revision leads to acceptable results. With a more sophisticated alignment algorithm, I expect that this problem will be reduced. Mainly, short sentences and fixed phrases work well, whereas long sentences cause more difficulties, also due to the still limited syntactic analysis.

5 CONCLUSION

A possible way for building an example based machine translation system was shown. All necessary steps were presented: beginning with syntactic analysis, explaining the preprocessing of the sample corpus and showing how the actual translation works. This includes how the input is split into parts, how these parts are translated using the sample corpus and how a target sentence is generated from them.

Although more in-depth evaluation still needs to be done, the first results are encouraging. Of course, there is much room for future research. E.g., the weights for the distance values could be determined automatically, the alignment algorithm could be improved to handle more sentence pairs correctly and a module filtering out identical translations in the example corpus should be introduced to reduce space consumption.

6 REFERENCES