Efficient Search for Interactive Statistical Machine Translation

Franz Josef Och[1] and **Richard Zens** and **Hermann Ney**
Chair of Computer Science VI
RWTH Aachen - University of Technology
{och,zens,ney}@cs.rwth-aachen.de

Abstract

The goal of interactive machine translation is to improve the productivity of human translators. An interactive machine translation system operates as follows: the automatic system proposes a translation. Now, the human user has two options: to accept the suggestion or to correct it. During the post-editing process, the human user is assisted by the interactive system in the following way: the system suggests an extension of the current translation prefix. Then, the user either accepts this extension (completely or partially) or ignores it. The two most important factors of such an interactive system are the quality of the proposed extensions and the response time. Here, we will use a fully fledged translation system to ensure the quality of the proposed extensions. To achieve fast response times, we will use word hypotheses graphs as an efficient search space representation. We will show results of our approach on the Verbmobil task and on the Canadian Hansards task.

1 Introduction

Current machine translation technology is not able to guarantee high quality translations for large domains. Hence, in many applications, post-editing of the machine translation output is necessary. In such an environment, the main goal of the machine translation system is not to produce translations that are understandable for an inexperienced recipient but to support a professional human post-editor.

Typically, a better quality of the produced machine translation text yields a reduced post-editing effort. From an application point of view, many additional aspects have to be considered: the user interface, the used formats and the additional support tools such as lexicons, terminological databases or translation memories.

The concept of *interactive machine translation*, first suggested by (Foster et al., 1996), finds a very natural implementation in the framework of statistical machine translation. In interactive machine translation, the basic idea is to provide an environment to a human translator that interactively reacts upon the input as the user writes or corrects the translation. In such an approach, the system suggests an extension of a sentence that the human user either accepts or ignores. An implementation of such a tool was performed in the TransType project (Foster et al., 1996; Foster et al., 1997; Langlais et al., 2000).

The user interface of the TransType system combines a machine translation system and a text editor into a single application. The human translator types the translation of a given source text. For each prefix of a word, the machine translation system computes the most probable extension of this word and presents this to the user. The human translator either accepts this translation by press-

[1]The author is now affiliated with the Information Science Institute, University of Southern California, och@isi.edu.

ing a certain key or ignores the suggestion and continues typing.

Rather than single-word predictions, as in the TransType approach, it is preferable that the suggested extension consists of multiple words or whole phrases. Ideally, the whole sentence should be suggested completely and the human translator should have the freedom to accept any prefix of the suggested translation.

In the following, we will first describe the problem from a statistical point of view. For the resulting decision rule, we will describe efficient approximations based on word hypotheses graphs. Afterwards, we will present some results. Finally, we will describe the implemented prototype system.

2 Statistical Machine Translation

We are given a source language ('French') sentence $f_1^J = f_1 \ldots f_j \ldots f_J$, which is to be translated into a target language ('English') sentence $e_1^I = e_1 \ldots e_i \ldots e_I$. Among all possible target language sentences, we will choose the sentence of unknown length I with the highest probability:

$$\hat{e}_1^I = \operatorname*{argmax}_{I,e_1^I} \{Pr(e_1^I|f_1^J)\} \quad (1)$$

$$= \operatorname*{argmax}_{I,e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \quad (2)$$

The decomposition into two knowledge sources in Eq. 2 is the so-called source-channel approach to statistical machine translation (Brown et al., 1990). It allows an independent modeling of target language model $Pr(e_1^I)$ and translation model $Pr(f_1^J|e_1^I)$. The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence. The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. Here, we maximize over all possible target language sentences.

3 Interactive Machine Translation

In a statistical approach, the problem of finding an extension e_{i+1}^I of a given prefix e_1^i can be described by constraining the search to those sentences e_1^I that contain e_1^i as prefix. So, we maximize over all possible extensions e_{i+1}^I:

$$\hat{e}_{i+1}^I = \operatorname*{argmax}_{I,e_{i+1}^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \quad (3)$$

For simplicity, we formulated this equation on the level of whole words, but of course, the same method can also be applied at the character level.

In an interactive machine translation environment, we have to evaluate this quantity after every key-stroke of the human user and compute the corresponding extension. For the practicability of this approach, an efficient maximization in Eq. 3 is very important. For the human user, a response time larger than a fraction of a second is not acceptable. The search algorithms developed so far are not able to achieve this efficiency without an unacceptable amount of search errors. The one we will use usually takes a few seconds per sentence. Hence, we have to perform certain simplifications making the search problem feasible.

Our solution is to precompute a subset of possible word sequences. The search in Eq. 3 is then constrained to this set of hypotheses. As data structure for efficiently representing the set of possible word sequences, we use word hypotheses graphs (Ney and Aubert, 1994; Ueffing et al., 2002) .

4 Alignment Templates

As specific machine translation method, we use the alignment template approach (Och et al., 1999). The key elements of this approach are the *alignment templates*, which are pairs of source and target language phrases together with an alignment between the words within the phrases. The advantage of the alignment template approach compared to single word-based statistical translation models is that word context and local changes in word order are explicitly considered.

The alignment template model refines the translation probability $Pr(f_1^J|e_1^I)$ by introducing two hidden variables z_1^K and a_1^K for the K alignment templates and the alignment of the alignment tem-

plates:

$$Pr(f_1^J|e_1^I) = \sum_{z_1^K, a_1^K} Pr(a_1^K|e_1^I) \cdot Pr(z_1^K|a_1^K, e_1^I) \cdot Pr(f_1^J|z_1^K, a_1^K, e_1^I)$$

Hence, we obtain three different probability distributions: $Pr(a_1^K|e_1^I)$, $Pr(z_1^K|a_1^K, e_1^I)$ and $Pr(f_1^J|z_1^K, a_1^K, e_1^I)$. Here, we omit a detailed description of modeling and training as this is not relevant for the subsequent exposition. For further details, see (Och et al., 1999).

5 Word Hypotheses Graphs

A word hypotheses graph is a directed acyclic graph $G = (V, E)$. It is a subset of the search graph and is computed as a byproduct of the search algorithm. Each node $n \in V$ corresponds to a partial translation hypothesis. Each edge $(n, n') \in E$ is annotated with both a target language word $e(n, n')$ and the associated extension probability $p(n, n')$ of language and translation model. The word hypotheses graph is constructed in such a way that the extension probabilities only depend on the two adjacent nodes. So, these probabilities are independent of the considered path through the graph. For simplicity, we assume that there exists exactly one goal and one start node. For a more detailed description of word hypotheses graphs, see (Ueffing et al., 2002). An example of a simplified word hypotheses graph is shown in Fig. 1 for the German source sentence "was hast du gesagt?". The English reference translation is "what did you say?".

For each node in the word hypotheses graph, the maximum probability path to reach the goal node is computed. This probability can be decomposed into the so-called forward probability $p(n)$, which is the maximum probability to reach the node n from the start node and the so-called backward probability $h(n)$, which is the maximum probability to reach the node n backwards from the goal node.

The backward probability $h(n)$ is an optimal heuristic function in the spirit of A* search. Having this information, we can compute efficiently for each node n in the graph the best successor node $S(n)$:

$$S(n) = \operatorname*{argmax}_{n':(n,n')\in E} \{p(n) \cdot p(n, n') \cdot h(n')\} \quad (4)$$

As each node n corresponds to a partial translation hypothesis e_1^i, the optimal extension of this prefix is obtained by:

$$\hat{e}_{i+1} = e(n, S(n)) \quad (5)$$

$$\hat{e}_{i+2} = e(S(n), S^2(n)) \quad (6)$$

$$\ldots$$

$$\hat{e}_{i+k} = e(S^{k-1}(n), S^k(n)) \quad (7)$$

Hence, the function S provides the optimal word sequence in a time complexity linear to the number of words in the extension.

Yet, as the word hypotheses graph contains only a subset of the possible word sequences, we might face the problem that the prefix path is not part of the word hypotheses graph. To avoid this problem, we perform a tolerant search in the word hypotheses graph. We select the set of nodes that correspond to word sequences with minimum Levenshtein distance (edit distance) to the given prefix. This can be computed by a straightforward extension of the normal Levenshtein algorithm for word hypotheses graphs. From this set of nodes, we choose the one with maximum probability and compute the extension according to Eq. 4. Because of this approximation, the suggested translation extension might contain words that are already part of the translation prefix.

6 Evaluation Criterion

As evaluation criterion, we use the key-stroke ratio (KSR), which is the ratio of the number of key-strokes needed to produce the single reference translation using the interactive translation system divided by the number of key-strokes needed to simply type the reference translation. We make the simplifying assumption that the user can accept an arbitrary length of the proposed extension using a single key-stroke. Hence, a key-stroke ratio of 1 means that the system was never able to suggest a correct extension. A very small key-stroke ratio means that the suggested extensions are often correct. This value gives an indication about the possible effective gain that can be achieved if this in-

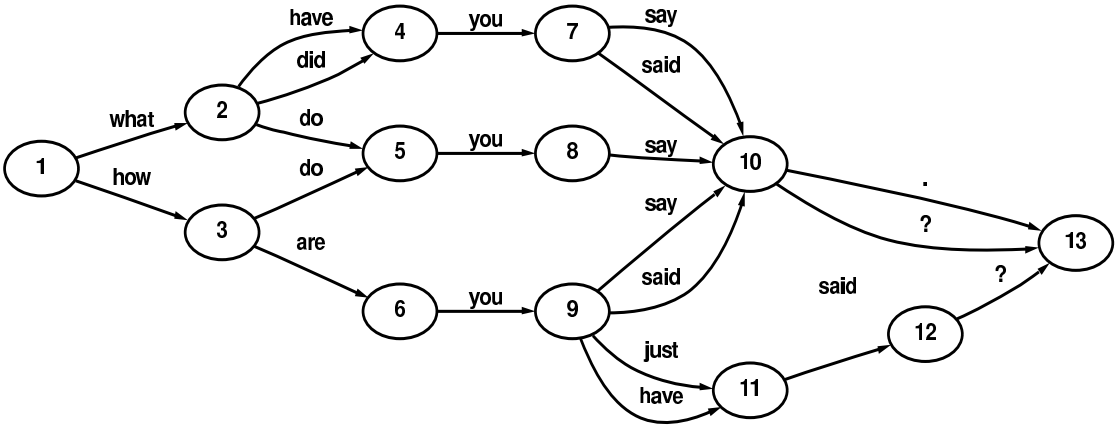

Figure 1: Example of a word hypotheses graph for the German source sentence "was hast du gesagt?" (English reference translation: "what did you say?").

teractive translation system is used in a real translation task. On the one hand, the key-stroke ratio is very optimistic with respect to the efficiency gain of the user. On the other hand, it is a well-defined objective criterion that we expect to be well correlated to a more user-centered evaluation criterion.

A simplified example is shown in Tab. 1. We manually selected paths in the word hypotheses graph (Fig. 1) to illustrate the interaction with the system. In practice, the system should translate this short sentence correctly without any user interaction. The reference translation is "what did you say ?" and the first suggestion of the system is "what do you say ?". So, the user accepts the prefix "what d" with one key-stroke (denoted with a "#") and then enters the correct character "i". The next suggestion of the system is "what did you said ?". Now, the user accepts the prefix "what did you sa" and then types the character "y". Finally, the system suggests the correct translation the user simply accepts. Overall, the user needed 5 key-strokes to produce the reference translation with the interactive translation system. Simply typing the reference translation would take 19 key-strokes (including blanks and a return at the end). So, the key-stroke ratio is $5/19 = 26.3\%$.

Table 1: Example of the post-editing process.

step no.	source	`was hast du gesagt ?`
	reference	`what did you say ?`
1	prefix	
	extension	`what do you say ?`
	user	`#i`
2	prefix	`what di`
	extension	`d you said ?`
	user	`#y`
3	prefix	`what did you say`
	extension	`?`
	user	`#`

7 Results

7.1 Verbmobil

The first task, we present results on, is the VERBMOBIL task (Wahlster, 2000). The domain of this corpus is appointment scheduling, travel planning, and hotel reservation. It consists of transcriptions of spontaneous speech. Table 2 shows the corpus statistics of this corpus.

Table 3 shows the resulting key-stroke ratio and the average extension time for various word hypotheses graph densities (i.e. the number of edges per source word). The table shows the effect of both single-word extensions and whole-sentence extensions.

We see a strong correlation between the word hypotheses graph density and the response time.

Table 2: Statistics of training and test corpus for Verbmobil (PP=perplexity).

		German	English
Train	Sentences	58 073	
	Words	519 523	549 921
	Vocabulary	7 939	4 672
	Singletons	3 453	1 698
Test	Sentences	251	
	Words	2 628	2 871
	Trigram PP	-	30.5

Table 3: Verbmobil: key-stroke ratio (KSR) and average extension time for various word hypotheses graph densities (WGD).

WGD	extension type			
	single-word		full sentence	
	time [s]	KSR [%]	time [s]	KSR [%]
5	0.003	54.3	0.003	41.7
14	0.008	47.6	0.008	32.3
32	0.014	45.7	0.015	29.6
77	0.022	44.6	0.025	28.1
188	0.034	43.8	0.038	27.0
453	0.050	43.0	0.058	25.7
1030	0.071	42.3	0.091	25.7
2107	0.106	42.0	0.143	25.0
3892	0.161	41.9	0.226	25.1
6513	0.235	41.7	0.345	24.7
10064	0.333	41.6	0.505	24.5

When using a larger word hypotheses graph, a considerably larger amount of time is needed to search for the optimal extension. On the other hand, there is a reduction of the KSR: in the case of single-word extensions, the KSR improves from 54.3% and 0.003 seconds per extension to 41.6% and 0.333 seconds per extension. Significantly better results are obtained by performing whole-sentence extensions. Here, the KSR improves from 41.7% and 0.003 seconds per extension to 24.5% and 0.505 seconds per extension.

Table 4: Statistics of training and test corpus for the Canadian Hansards task (PP=perplexity).

		French	English
Train	Sentences	1.5M	
	Words	24M	22M
	Vocabulary	100 269	78 332
	Singletons	40 199	31 319
Test	Sentences	200	
	Words	2 124	2 246
	Trigram PP	–	180.5

7.2 Canadian Hansards

Additional experiments were carried out on the Canadian Hansards task. This task contains the proceedings of the Canadian parliament, which are kept by law in both French and English. About 3 million parallel sentences of this bilingual data have been made available by the Linguistic Data Consortium (LDC). Here, we use a subset of the data containing only sentences with a maximum length of 30 words. Table 4 shows the training and test corpus statistics.

Table 5 shows the resulting key-stroke ratio and the average extension time for various word hypotheses graph densities. Again, we show the effect of both single-word extensions and whole-sentence extensions.

The results are similar to the Verbmobil task: by using a larger word hypotheses graph, a considerably larger amount of time is needed to search the word hypotheses graph, but on the other hand there is an improvement of the KSR: in the case of single-word extensions, the KSR improves from 62.9% and 0.003 seconds per extension to 50.3% and 0.436 seconds per extension. As for the Verbmobil task, significantly better results are obtained by performing whole-sentence extensions. Here, the KSR improves from 46.3% and 0.002 seconds per extension to 33.1% and 0.556 seconds per extension.

Regarding the experiments carried out on both tasks, we conclude that the set of possible candidate translations can be indeed represented by word hypotheses graphs. In addition, we conclude that whole-sentence extensions give significantly better results than single-word extensions.

Table 5: Hansards: key-stroke ratio (KSR) and average extension time for various word hypotheses graph densities (WGD).

WGD	extension type			
	single-word		full sentence	
	time [s]	KSR [%]	time [s]	KSR [%]
11	0.003	62.9	0.002	46.3
22	0.009	58.0	0.009	40.9
83	0.028	54.2	0.028	36.6
363	0.059	52.9	0.061	35.8
1306	0.104	52.0	0.113	34.9
3673	0.172	51.3	0.194	34.0
8592	0.274	50.8	0.329	33.5
17301	0.436	50.3	0.556	33.1

8 Prototype System

In the following, we describe how the presented method has been used to build an operational prototype for interactive translation. This prototype has been build as part of the EU project TransType 2 (IST-2001-32091). It allows an effective interaction between the human translator and the machine translation system. The prototype has the following key properties:

- The system uses the alignment template approach described in section 4 as translation engine.
- It allows the machine translation output to be interactively post-edited. The system suggests a full-sentence extension of the current translation prefix. The user either accepts the complete suggestion or a certain prefix.
- The human translator is able to obtain a list of alternative words at a specific position in the sentence. This helps the human translator to find alternative translations.
- Since the system is based on the statistical approach, it can learn from existing sample translations. Therefore, it adapts to very specific domains without much human intervention. Unlike systems based on translation memories, the system is able to provide suggestions also for sentences that have not been seen in the bilingual translation examples.
- The system can also learn interactively from those sentences that have been corrected or accepted by the user. The user may request that a specific set of sentences be added to the knowledge base. A major aim of this feature is an improved user acceptability as the machine translation environment is able to adapt rapidly and easily to a new vocabulary.

The developed system seems to have advantages over currently used machine translation or translation memory environments as it combines important concepts from these areas into a single application. The two major advantages are the ability to suggest full-sentence extensions and the ability to learn interactively from user corrections.

The system is implemented as a client–server application. The server performs the actual translations as well as all time-consuming operations such as computing the extensions. The client includes only the user interface and can therefore run on a small computer. Client and server are connected via Internet or Intranet.

There is ongoing research to experimentally study the productivity gain of such a system for professional human translators.

9 Related Work

As already mentioned, previous work towards interactive machine translation has been carried out in the TransType project (Foster et al., 1996; Foster et al., 1997; Langlais et al., 2000).

In (Foster et al., 2002) a so-called "user model" has been introduced to maximize the expected benefit of the human translator. This user model consists of two components. The first component models the benefit of a certain extension. The second component models the acceptance probability of this extension. The user model is used to determine the length of the proposed extension measured in characters.

The resulting decision rule is more centered on the human user than the one in Eq. 3. It takes into account, e.g., the time the user needs to read the extension (at least approximatively).

In principle, the decision rule in Eq. 3 can be extended by such a user model. In (Foster et al., 2002) the assumption is made that "the user edits only by erasing wrong character from the end of a proposal". The approach in this paper is different in that the user works from left to right by either accepting or correcting the proposed translation. Therefore, in our approach, we would have to modify the details of the user model.

An additional difference is the used translation engine: in (Foster et al., 2002) a simple translation model is chosen for efficiency reasons, namely a maximum entropy version of IBM2. Here, we use a fully fledged translation model and deal with the efficiency problem by using word hypotheses graphs.

10 Conclusions

We have suggested an interactive machine translation environment for computer assisted translation. It assists the human user by interactively reacting upon his/her input. The system suggests full-sentence extensions of the current translation prefix. The human user can accept any prefix of this extension.

We have used a fully fledged translation system, namely the alignment template approach, to produce high quality extensions. Word hypotheses graphs have been used to allow an efficient search for the optimal extension. Using this method, the amount of key-strokes needed to produce the reference translation reduces significantly.

Additional optimizations of the word hypotheses graphs might improve the efficiency of the search. E.g., forward-backward pruning (Sixtus and Ortmanns, 1999) could be used to reduce the word hypotheses graph density. Further improvements could be achieved by incorporating a more user-centered cost function like the user model in (Foster et al., 2002). To answer the question of how long the extension should be, a good confidence measure could be useful (Wessel et al., 2001).

11 Acknowledgement

This work has been partially funded by the EU project TransType 2, IST-2001-32091.

References

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.

G. Foster, P. Isabelle, and P. Plamondon. 1996. Word completion: A first step toward target-text mediated IMT. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pages 394–399, Copenhagen, Denmark, August.

G. Foster, P. Isabelle, and P. Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12(1):175–194.

G. Foster, P. Langlais, and G. Lapalme. 2002. User-friendly text prediction for translators. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 46–51, Philadelphia, July.

P. Langlais, G. Foster, and G. Lapalme. 2000. TransType: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems*, pages 46–51, Seattle, Wash., May.

H. Ney and X. Aubert. 1994. A word graph algorithm for large vocabulary continuous speech recognition. In *Proc. Int. Conf. on Spoken Language Processing*, pages 1355–1358, Yokohama, Japan, September.

F. J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June.

A. Sixtus and S. Ortmanns. 1999. High quality word graphs using forward-backward pruning. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 593–596, Phoenix, AZ, USA, March.

N. Ueffing, F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. Conf. on Empirical Methods for Natural Language Processing*, pages 156–163, Philadelphia, PA, July.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, July.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney. 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, March.