Comparison of Generation Strategies for Interactive Machine Translation

Oliver Bender, Saša Hasan, David Vilar, Richard Zens, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{bender,hasan,vilar,zens,ney}@cs.rwth-aachen.de

Abstract. Fully automatic translations are far from being perfect. Non-grammatical sentences are often produced by automatic systems and there is even no guarantee that the meaning of the sentence is preserved. Nevertheless, automatic translation systems can be used to help human translators to produce high-quality translations. This is the goal of the TransType2 project, where an interactive translation tool is being developed that suggests, in real time, possible completions for the sentences that the human translator is typing. This leads to a modification of the generation strategy of the translation system, as now we are looking for the best translation of the given source sentence that is compatible with the prefix. In order to remain within the tight response time constraints of such a system, some simplifications have to be done. In this paper, we review possible generation strategies for an interactive statistical machine translation system and analyze what is the loss in performance when strict time constraints have to be met. Experiments are performed on the Spanish-English and German-English Xerox corpora, which consist of the translation of technical manuals, and the results show that the real time generation strategy causes only a small performance degradation.

1 Introduction

Although great progress has been made in the field of automatic translation in the last years, the produced translations are far from being perfect. Apart for very limited domains, no current state-of-the-art system can be directly used for real life applications. The produced sentences often contain grammatical errors and the preservation of the meaning is not even always achieved. Therefore a manual post-processing of the texts has to be done.

The concept of *interactive machine translation* already has a long history, and the first systems appeared in the end of the 1960's. However in most of these systems the user doesn't have a direct control over the translation process, and most of the user interaction is reduced to performing source language disambiguation on demand. The approach we center on in this work was first suggested by (Foster et al., 1996) and an implementation was carried out in the TransType project (Langlais et al., 2000). In such an environment, human translators interact with a translation system that acts as an assistance tool and dynamically provides a list of translations that best complete the part of the source sentence already translated. Further refinements were presented in the TransType2 project (SchlumbergerSema S.A. et al., 2001).

The work presented in this paper deals with generation strategies for interactive (statistical) machine translation systems. Clearly, the best approach would be to start a new search for every given prefix. However, in these kind of systems, response time is a crucial factor for a human translator, as delays higher than a fraction of a second are not acceptable. With today's algorithms and available computing power, these time restrictions can not be met when doing a whole new search for each prefix, so the performance achieved with this strategy will be an upper bound of the performance we get in the real system. We will present an efficient generation strategy and compare its capability with this upper bound.

The remainder of the paper is organized as follows: first we will describe the concept of interactive machine translation from a statistical point of view. Next, we will discuss the two generation strategies mentioned above and how they can be successfully combined. After that, experimental results comparing the generation strategies are presented and conclusions are drawn.

2 Interactive Machine Translation

In this section we will briefly review the statistical framework for translation our system builds on. In statistical machine translation we are given a source sentence f_1^J which is translated into a target sentence e_1^I which maximizes the posterior probability

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\}. \tag{1}$$

Applying Bayes' Rule, we can modify this equation in order to introduce additional knowledge sources

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\}. \tag{2}$$

The target language model $Pr(e_1^I)$ describes the well-formedness of the target language sentence. The translation model $Pr(f_1^J|e_1^I)$ 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.

In interactive machine translation, we have to find an extension e_{i+1}^I for a given prefix e_1^i. Hence, we constrain the search to those sentences e_1^I which contain e_1^i as prefix:

$$\begin{aligned} \hat{e}_{i+1}^I = \underset{I, \tilde{e}_{i+1}^I}{\operatorname{argmax}} \Big\{ & Pr(\tilde{e}_{i+1}^I | e_1^i) \cdot \\ & Pr(f_1^J | e_1^i, \tilde{e}_{i+1}^I) \Big\}. \end{aligned} \tag{3}$$

Thus, we maximize over all possible extensions $\tilde{e}_{i+1}^I$. For simplicity, this equation is formulated on the word level. We do not include the case where the prefix contains the first characters of the word e_{i+1}. In that case, we have to optimize over all target language words e_{i+1} that have the same word prefix. In the actual implementation, the method is applied on the character level, and the search for an extension is performed after each keystroke of the human translator.

The crucial factor is an efficient maximization of Eq. 3, because human translators will only accept response times of fractions of a second. Using state-of-the-art search algorithms this is not achievable without putting up with an unacceptable amount of search errors. To overcome this problem, we can compute a word graph which represents a subset of possible extensions (Ney and Aubert, 1994; Ueffing et al., 2002). The generation is then constrained to this set of extensions.

3 Phrase-based Approach

The base method we use in our translation system is the alignment template approach as described in (Och et al., 1999; Och and Ney, 2004). This approach uses the so-called *alignment templates*, which are pairs of source and target language phrases[1] together with the word alignment within the phrases. The alignment templates are introduced as hidden variables z_1^K when modelling the conditional translation probability $\Pr(f_1^J|e_1^I)$:

$$\begin{aligned} \Pr(f_1^J|e_1^I) = \sum_{z_1^K, a_1^K} & \Pr(a_1^K|e_1^I) \cdot \Pr(z_1^K|a_1^K, e_1^I) \cdot \\ & \Pr(f_1^J|z_1^K, a_1^K, e_1^I). \end{aligned} \tag{4}$$

In Equation (4), we introduce the additional hidden variables a_1^K that model the alignment of the alignment templates themselves. As smoothing, automatically trained word classes can be used, and additional costs can easily be introduced by using a log-linear model. More details of this approach can be found in the literature.

3.1 Generation

The generation of the best translation for a given source sentence f_1^J is carried out by producing the target sentence in a sequential order. At each step of the generation algorithm we maintain a set of active hypotheses and choose one of them for extension. A word of the target language is then added to the chosen hypothesis and its costs get updated. This kind of generation fits nicely into a dynamic programming framework, as hypotheses which are indistinguishable by both language and translation models (and that have covered the same source positions) can be recombined. The search space is however too big, and therefore pruning has to be done, which leads us to a beam search algorithm.

[1] In this context, phrases are simply sequences of words. No other linguistic meaning is required.

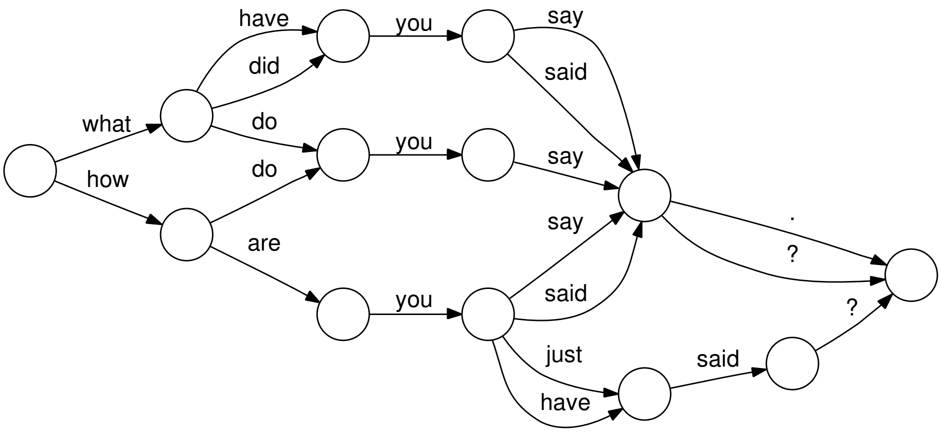

Figure 1. Example of word graph for the German sentence "was hast du gesagt?" (English reference translation: "what did you say?").

4 Interactive Generation

In order to find the completion for a given prefix, the set of generated hypotheses could be restricted to only those which exactly match the given prefix. However, as our probabilistic models are far from being perfect, this approach is too restrictive. Instead, we penalize the hypotheses by introducing an additional cost in the log-linear model for each word that does not match the prefix. If hypotheses that can generate the given prefix are present in the active set, those do not get any additional costs. In the pruning process, the incompatible hypotheses will be discarded while the correct ones will remain in the set. Of course the last "word" in the given prefix should be considered in a different way, as it itself can be a prefix of the next word. To ensure the extensions start with this word prefix, the comparison must be done at the character level. One might think about different costs for the mismatch of words within the prefix and for extensions which do not start with the given word prefix. If a word within the prefix can not be produced by the search algorithm, then it will obviously not be produced by any further search call. This kind of substitution error is less harmful for producing good hypotheses than unfitting extensions, and should therefore be penalized less.

Using this approach we can expect to obtain optimal results, as a new search is performed at each stage, and the information provided by the prefix is used to avoid search errors made in previous stages. However, the search process has a high computational cost and in the interactive systems the response time is a critical point. Therefore, this approach can normally not be used for practical application and some more time-efficient alternatives have to be found.

5 Interactive Generation with Word Graphs

In (Och et al., 2003), an efficient algorithm for interactive generation using word graphs was presented. A word graph is a weighted directed acyclic graph, in which each node represents a partial translation hypothesis and each edge is labeled with a word of the target sentence and is weighted according to the language and translation model scores. (Ueffing et al., 2002) give a more detailed description of word graphs and show how they can be easily produced as a sub-product of the search process. An example of a word graph is shown in Figure 2.

It is clear that each node in the word graph defines a prefix of a possible translation of the given source sentence. The main idea behind this approach is to find the node[2] that corresponds to the given prefix and generates the best completion starting from this node. This can be easily accomplished using a forward-backward algorithm.

[2] In the general case there can be more than one node that represents the same prefix. But with an appropriate determinization this case can be avoided.

As the word graph is a representation of a subset of the possible translations for a source sentence, it can happen that the given prefix can not be found in the word graph. In this case we look for the node with minimum edit distance to the prefix and select the completion path with best backward score. The algorithm for computing the edit distance between a string and such a graph is a straightforward extension of the Levenshtein algorithm for computing the distance between two strings.

The computational cost of this approach is much lower than that of the one presented in Section 4, as the whole search for the translation must be carried out only once, and the generated word graph can be reused for further completion requests. This is also, of course, its main limitation, as the word graph does not get automatically adapted to the new information provided by the prefix. This can be somehow alleviated by allowing a more flexible alignment of the generated sentences to the given prefix using the edit distance measure.

Another refinement can be added to the system. Usually, if the translation system was not able to find a completion in the generated word graph that is compatible with the last partial word in the prefix, the user has to type the whole completion. Instead, we now try to find the completion with highest probability using only the language model. This simple heuristic slightly increases the performance of the system, as words that were rejected in the pruning process can be recovered.

5.1 Combination of both strategies

In order to overcome the limitations of the generation with word graphs, we can try to combine both strategies. We start by generating a word graph for the translations of the given source sentence and then use it for searching for completions. If, at a certain point, we determine that the generated graph does not correspond with the prefix typed by the user, we generate a new word graph tailored to this prefix with the method described in Section 4. An important point is how to decide if the word graph should not be used any more and that a new one has to be generated. In our experiments we used a simple heuristic: if the last word in the prefix is not complete (i.e. the prefix does not end with a blank space) and the selected node in the word graph does not produce a completion for this word, the word graph gets regenerated. This simple criterion already leads to an improved performance over the standard search strategy using word graphs. What still has to be determined is if the response time of the system, increased by the overhead of regenerating the word graphs, remains acceptable for interactive use under real-life conditions. Off-line experiments seem to indicate that this is the case (see next section).

6 Experimental Results

6.1 Experimental Setup

The experiments were performed on the Spanish-English and English-German Xerox corpora, which consist of the translation of technical manuals. The corpora allocations are summarized in Table 1 and Table 2.

After training and optimization of the model scaling factors, the SMT engine (Bender et al., 2004) was used to translate the test corpus. The results using the standard evaluation measures for machine translation (word error rate, position-independent word error rate, BLEU score and NIST score) are shown in Table 3.

Using the same parameter settings, a simulation of the interactive mode was carried out. This simulation mode is described in (Och et al., 2003). The system with the same parameter settings was also successfully used by human translators to evaluate it under real-life conditions. Due to the high effort that human evaluations require, only the word-graph based generation strategy was tested. The response time of the system was adequate.

6.2 Evaluation Criterion

The evaluation is based on the so-called *keystroke ratio* (KSR) introduced in (Och et al., 2003), which divides the number of keystrokes needed to produce the single reference translation (using the interactive translation system) by the number of keystrokes needed to simply type the reference translation. Hence, a keystroke ratio of 1 means that the system was never able to suggest a correct extension, whereas a small keystroke ratio means that the produced extensions are often correct.

The KSR value is an indicator of the possible effective gain that can be achieved if this interac-

		Spanish	English
Train	Sentences	55 761	
	Running Words	752 166	666 700
	Running Words without Punct. Marks	693 017	608 254
	Vocabulary	16 362	13 541
	Singletons	5 046	3 725
Dev	Sentences	1 012	
	Running Words	15 999	14 352
	Running Words without Punct. Marks	14 745	13 071
	Vocabulary	1 793	1 648
	OOVs (running words)	95	55
	OOVs (in voc.)	67	36
Test	Sentences	1 125	
	Running Words	10 226	8 521
	Running Words without Punct. Marks	9 738	8 060
	Vocabulary	1 917	1 879
	OOVs (running words)	250	222
	OOVs (in voc.)	174	157

Table 1. Statistics of the Spanish-English Xerox (raw) corpus.

		German	English
Train	Sentences	49 376	
	Running Words	537 464	589 531
	Running Words without Punct. Marks	443 547	509 902
	Vocabulary	23 845	13 223
	Singletons	9 443	3 681
Dev	Sentences	964	
	Running Words	10 462	10 642
	Running Words without Punct. Marks	8 372	9 259
	Vocabulary	1 746	1 516
	OOVs (running words)	147	29
	OOVs (in voc.)	114	29
Test	Sentences	996	
	Running Words	11 704	12 298
	Running Words without Punct. Marks	9 711	10 656
	Vocabulary	2 179	1 838
	OOVs (running words)	485	141
	OOVs (in voc.)	310	95

Table 2. Statistics of the German-English Xerox (raw) corpus.

Language Pair	Evaluation Criteria			
	WER [%]	PER [%]	BLEU [%]	NIST
Spanish - English	40.2	34.4	57.2	8.7
English - Spanish	33.4	28.3	62.0	9.5
German - English	67.9	56.6	25.7	6.0
English - German	76.6	68.7	20.7	5.1

Table 3. Translation results for the Xerox (raw) Spanish-English and German-English task.

Generation Strategy	Translation Direction							
	Es - En		En - Es		Ge - En		En - Ge	
	time [ms]	KSR [%]	time [ms]	KSR [%]	time [ms]	KSR [%]	time [ms]	KSR [%]
interactive	2 489	20.3	3 283	21.8	2 747	38.8	2 661	39.1
combined	130	20.6	332	21.8	112	39.5	105	39.5
interactive with word graphs	17	21.1	13	21.7	25	39.9	28	40.0

Table 4. Average extension time and keystroke ratio (KSR) for the investigated generation strategies, both for the Spanish-English and the German-English Xerox (raw) task.

tive translation system is used in a real translation task. Although the keystroke ratio is very optimistic with respect to the efficiency gain of a user, it was shown in the TransType 2 project (SchlumbergerSema S.A. et al., 2001) that these measurements are correlated.

6.3 Results

Table 4 contains the average extension times and keystroke ratios for the investigated generation strategies, both for the Spanish-English and the German-English corpora, in both translation directions.

As can be seen, in nearly all translation directions the best performance in terms of keystroke ratio is achieved when carrying out a new search for every prefix. The only exception is for the English to Spanish direction, where the interactive search with word graphs is slightly better than the new search for every prefix. This can be due to the rich morphology of the Spanish language, where the correct form of some words can not be generated by the search procedure and thus, the flexibility provided by the use of the Levenshtein distance when searching allows for a better keystroke ratio. On the other hand, the average extension time for full search is far from being acceptable for real translation tasks.

The values for the system that uses the interactive search with word graphs (the one used in the human evaluation) uses the same parameters as the other systems, but the beam size was further reduced, in order to get a better response time. The average extension time gets significantly reduced while the keystroke ration increases only slightly, about 7% relative in the worst case (direction German to English).

Figure 2 shows the keystroke ratio as a function of the average extension time (controlled varying the size of the beam). As expected, the best keystroke ratio values are obtained at the expense of a high extension time. Nevertheless the interactive search with word graphs achieves already adequate results for low extension times, i.e. this strategy fits the tight response time constraints of real-life systems. In addition, Table 5 shows the different word graph densities associated with different extension times.

The combined generation strategy helps alleviating the performance loss and in all the cases pro-

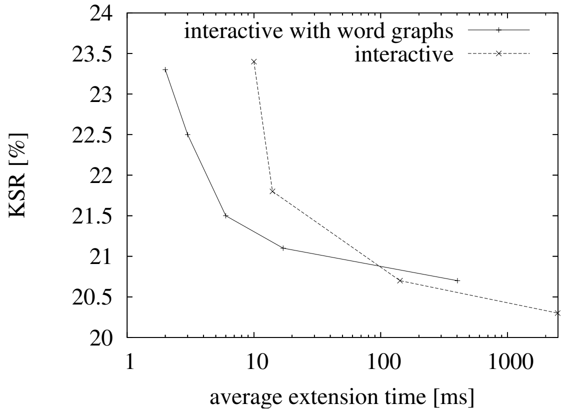

Figure 2. Keystroke ratio (KSR) as a function of the average extension time for the interactive and the interactive with word graphs generation strategy for the Spanish-English Xerox (raw) task.

WGD	KSR [%]	time [ms]
4	23.3	2
6	22.5	3
57	21.5	6
234	21.1	17
3400	20.7	403

Table 5. Keystroke ratio (KSR) and average extension time for different word graph densities (WGD) for the Spanish-English Xerox (raw) task.

vides a keystroke ratio value between the one of the whole search and the keystroke ratio of the search with word graphs. The average completion time seems to indicate that this strategy could also be used in the interactive environment, but this has still to be tested under real-life conditions[3].

Table 6 gives an example of a sentence from the English-German test corpus that is translated using the interactive generation with word graphs and by applying the pure interactive generation (full search) strategy. The part of the translation that has been accepted by the user is taken as prefix for the search for the next extension. We see that the correct result is obtained much faster with the full search method: only four steps of system-user interaction are necessary instead of seven. The benefit is due to the fact that the initial word graph does not contain the word “DocuColor”. Hence, the correct extension can not be found directly but has to be produced more or less character by character using the language model heuristic. In contrast, the interactive strategy performs a new search given the prefix “Komponenten des Do” and is then able to produce the correct extension in one step. The number of keystrokes to type the reference decreases from 11 to 6 (with 38 reference characters); KSR decreases from 28.9% to 15.8%. This benefit can also be achieved using the combined generation strategy. Here, a new word graph is computed for the given prefix “Komponenten des Do”, resulting in the same one step production of the correct extension.

7 Conclusion

In this paper, we have shown how the generation strategy for a state-of-the-art statistical machine translation system can be adapted for use in an interactive environment. The first approach consists in outputting only the translations compatible with the given prefix. Because this approach needs to perform a new search after each keystroke of the user, the real-time constraints of an interactive machine translation system do not allow to use this generation algorithm in practice. In this paper, we have reviewed an efficient generation process which generates a word graph for a given source sentence and looks for completions of the prefixes within this word graph. The performance of the system degrades slightly but the search is performed in a much more efficient way. Furthermore, a combination of both strategies has been proposed which improves the translation qualities and offline experiments seem to show that the response time can be adequate for real-time responsiveness, although this has not been tested yet under real-life conditions.

Acknowledgement

This work was partly funded by the European Union under the RTD project TransType2 (IST–2001–32091), and by the Deutsche Forschungsgemeinschaft (DFG) under the project “Statistische Textübersetzung” (Ne572/5).

[3] Consider also the subjective impression of a “long” waiting time only when selecting a new sentence to translate and then nearly instant completions (word-graph based search) against “random” waiting times when typing the translation (combined strategy).

SOURCE		Component parts of the DocuColor 12 Printer
REFERENCE		Komponenten des DocuColor 12 Druckers
interactive generation with word graphs	prefix	
	extension	Komponenten der DocuColor 12
	prefix	Komponenten des
	extension	
	prefix	Komponenten des D
	extension	er DocuColor 12
	prefix	Komponenten des Do
	extension	cument
	prefix	Komponenten des DocuC
	extension	olor
	prefix	Komponenten des DocuColor 1
	extension	2
	prefix	Komponenten des DocuColor 12 D
	extension	ruckers
interactive generation	prefix	
	extension	Komponenten der DocuColor 12
	prefix	Komponenten des
	extension	Komponenten der DocuColor 12
	prefix	Komponenten des D
	extension	er DocuColor 12
	prefix	Komponenten des Do
	extension	cuColor 12 Druckers

Table 6. Comparison of the interactive generation with word graphs and the interactive generation strategy for an example from the English-German test set; simulated interactive mode.

8 References

O. Bender, R. Zens, E. Matusov, and H. Ney. 2004. Alignment Templates: the RWTH SMT System. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 79–84, Kyoto, Japan, September.

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*, pp. 394–399, Copenhagen, Denmark, August.

P. Langlais, G. Foster, and G. Lapalme. 2000. TransType: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems*, pp. 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*, pp. 1355–1358, Yokohama, Japan, September.

F.J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

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

F.J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *EACL03: 10th Conf. of the Europ. Chapter of the Association for Computational Linguistics*, pp. 387–393, Budapest, Hungary, April.

SchlumbergerSema S.A., Intituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen - Lehrstuhl für Informatik VI, Recherche Appliquée en Linguistique Informatique Laboratory - University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe. 2001. TT2. TransType2 - computer assisted translation. Project technical annex.

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