Translation with Cascaded Finite State Transducers

Stephan Vogel and Hermann Ney
Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen – University of Technology
D-52056 Aachen, Germany
vogel@informatik.rwth-aachen.de

Abstract

In this paper we discuss the use of cascaded finite state transducers for machine translation. A number of small, dedicated transducers is applied to convert sentence pairs from a bilingual corpus into generalized translation patterns. These patterns, together with the transducers are then used as a hierarchical translation memory for fully automatic translation. Results on the German–English VERBMOBIL corpus are given.

1 Introduction

Corpus based approaches to automatic translation come in a number of different flavors. In the simplest form, translations are stored and reused for the translation of new input. This approach, known as translation memory, example-based or case-based translation, can work on the word level as well as on structured examples as they are generated during analysis and generation in more grammar-based translation paradigms (Kitano, 1993; Brown, 1996).

Finite state transducers, which can be learned from bilingual corpora, have been proposed for automatic translation (Amengual et al., 2000), as have been bilingual stochastic grammars (Wu, 1996). Statistical approaches (Wang and Waibel, 1997; Och et al., 1999) also fall into the category of corpus based approaches.

In this paper, a translation method is proposed which is based on the very same principles as the aforementioned approaches. One difference is, that not a fully automatic training of the translation model is performed. Rather, a number of special purpose transducers are hand-crafted and used then at two points. First, to convert the bilingual training corpus into a translation memory containing translation patterns rather than merely sentence pairs, and which is itself used as a transducer in the translation process. Second, when new sentences are to be translated, the transducers are applied to transform the input sentence into one or many possible target sentences the best of which, according to some scoring scheme, is selected as the translation.

In the next section, the construction of the transducers and the translation memory is outlined. Then, the application of the transducers for the translation of new sentences is described. In the last section the results of some translation experiments are given.

2 The Transducers

2.1 Overview

A finite state transducer (FST) is a finite state device which reads symbols from one channel and outputs a stream of symbols to a second channel. So, a FST can be depicted as a transition net with edges and nodes, where the nodes represent the states and the edges the possible state transitions. The edges are labelled with an input symbol and an output string, which may be the empty words of the two vocabularies. The final states can produce additional output.

We want to construct transducers for automatic machine translation from a given bilingual corpus. In fact, a collection of sentence pairs can be viewed as a trivial transducer, where each sentence pair is represented by a distinct line of nodes connected by edges labeled with the source sentence words and the target sentence emitted from the final state. This can be easily transformed into a tree transducer by building a prefix tree over the source sentences.

In (Amengual et al., 2000) a method is given to propagate prefixes of the translations towards the root of such a tree transducer and to coalesce states to gain generalization power. We choose here a different route to generalization by using an approach similar to the one used for chunk parsing, where a cascade of FST is applied (Abney, 1997). Each transducer, defined by a set of regular-expression patterns, reads part of the input sentence and writes a stream of category labels, which form, together with the unanalyzed parts of the sentence, the input to the next transducer in the cascade.

Our approach differs from the aforementioned chunk parsing in that an analyzed sequence of words is not replaced by the category label but is kept as a parallel option for transducers applied at a later stage. How this leads to the construction of a translation graph will be explained in Section 3.

For translation, not only the analysis of the source sentence is required but also the generation of the target sentence. This can be achieved if the transducers write category labels as well as translations to the output channel.

We allow for more than one translation for a given input sequence. This raises the question of how to select one translation over the others. Some kind of scoring is required, a point we will return to in section 2.3.

To summarize: each transducer is given as a set of quadruples of the form: [label # source pattern # target pattern # score]. At runtime these patterns are stored in a prefix tree with respect to the source patterns. We write the labels at first position as these translations patterns can be used in the reverse direction, i.e. from target language to source language. In section 2.4 this property is used to convert a bilingual corpus into a set of translation patterns which are formulated in terms of words and category labels. It also shows the structural identity to bilingual grammars as used in (Wu, 1996).

2.2 Construction of the Transducers

Most of the transducers are customized towards the domain for which the translation system is developed. In the VERBMOBIL Corpus, which is used for the experiments, time and date expressions are very prominent. To translate those expressions, a small grammar has been developed and coded as finite state transducer. Actually, two transducers are used. On the first level, words are replaced by labels, like DAYOFWEEK = {Montag/Monday, Dienstag/Tuesday, ...}. On the second level, these labels together with labeled numbers (ordinal, cardinal, fractions) from the number transducer are used to form complex time and date expressions. Some examples are given in Table 1.

All in all we use currently seven of those dedicated transducers: names (persons, towns, places, events, etc), spelling sequences (e.g. 'D A double L'), numbers (ordinal, cardinal, fractions, etc.), simple time and date expressions, compound time and date expressions, part-of-speech tagging, grammar (noun phrases, verb phrases). The relationship between these different transducers is depicted in figure 1. The arrows indicate that category labels introduces by one transducer are used by another transducer.

The division into these transducers is mainly a conceptual one. The five base level transducers could be coalesced into one transducer. Actually, this is done at runtime for efficiency. However, to keep them appart at construction time gives more flexibility. For example, while for a closed vocabulary in a speech translation task these transducers boil down to simple substitution list an open vocabulary task will require a more elaborate approach to proper name spotting or handling of numbers.

The part-of-speech transducer has been constructed semi-automatically. A tagger was

Table 1: Compound date translation patterns.

TIME	# um NUM_ORD Uhr	# at NUM_ORD o'clock	# -0.7
PERIOD	# NUM_CARD bis NUM_CARD	# NUM_CARD till NUM_CARD	# -0.7
PERIOD	# NUM_ORD Monate lang	# for NUM_ORD months	# -3.0
DATE_DAY	# am DAY_OF_WEEK	# on DAY_OF_WEEK	# -2
DATE	# in der NUM_CARD Woche	# in the NUM_CARD week	# -0.7
DATE	# Anfang MONTH	# beginning of MONTH	# -0.7
DATE	# DATE bis zum DATE	# from DATE till DATE	# -0.7

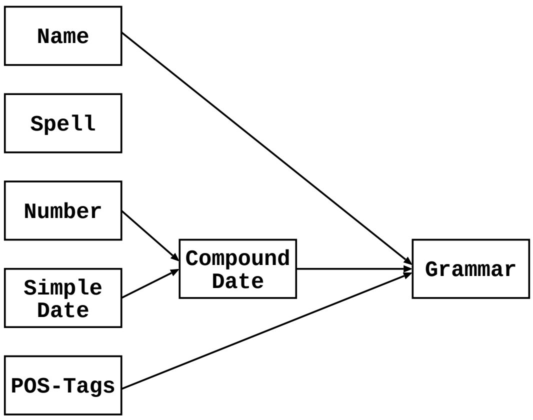

Figure 1: Hierarchy of transducers.

used to get a word – POS tag list. This was combined with an automatically generated translation lexicon (Och et al., 1999) to produce a list of label – word – translation patterns. This was then manually corrected and augmented where necessary.

Ideally, one would like to have a common tagset for both source and target language. If this is not available an alternative is to use a tagset for one language and induce via the word to word correspondences a tagging for the second language. This is the approach taken in this study. As tagset we use the Stuttgart-Tübinger tagset for German (Schiller et al., 1995).

Finally, a small bilingual grammar based on POS tags has been crafted manually. The purpose of the grammar is twofold: First, improving generalization by recognizing simple noun and prepositional phrases. Second, to handle the different word ordering in source and target language, especially in the verb phrases.

2.3 Scoring

The scores attached to the translation patterns can be viewed as a kind of translation scores. In the current implementation a rather crude heuristics together with some manual tuning in the grammar transducer is applied. The idea is to give preference to longer translation patterns as they take more context into account and encode word reordering in an explicit manner. So, for simple and compound translation patterns the score is exponential to the length of the source pattern. The scores are negative by convention: not translating a word gives zero cost, translating it gives a benefit, i.e. negative costs.

2.4 Bilingual Labeling

The sentence pairs in the bilingual training corpus could be used directly as a simple translation memory. However, to improve the coverage on unseen data, these segments are transformed into translation patterns containing category labels. For each transducer taken from the complete cascade – as given in Figure 1 – the transducers are applied to both, the source and the target sentences of the bilingual training corpus (Vogel and Ney, 2000). Those sentence pairs where number and types of category labels in source and target sentence match each other are selected into the database of compound translation patterns. Table 2 shows examples of some translation patterns which resulted from bilingual labeling.

3 The Translation Process

The working of the transducers is best described as the construction of a translation

Table 2: Compound translation patterns.

CTP # DATE ginge es wieder	# DATE it is possible again	# -4.6
CTP # NAME SURNAME am Apparat	# this is NAME SURNAME speaking	# -4.6
CTP # NP dauert DATE	# NP takes DATE	# -3.3
CTP # nehmen PPER NP DATE	# let PPER take NP DATE	# -4.6

graph. That is to say, the sentence to be translated is viewed as a graph which is traversed from left to right. For each matching source pattern, as stored in the transducers, a new edge is added to the graph. The edge is labeled with the category label of the translation pattern. The translation and the translation score are attached to the edge. In this way a translation graph is constructed. In those cases, where a source pattern has several translations, one edge for each translation is added to the graph. One advantage of this approach is that it can be applied to perform translation on word lattices as generated by speech recognition systems without any modifications.

The left–right traversal of the graph is organized in such a way that all paths are traversed in parallel and the patterns stored in the transducer are matched synchronously. For each node n and each edge e leading to that node all patterns in the transducer starting with the word or category label of e are attached to n. This gives a number of hypotheses describing partially matching patterns. Already started hypotheses are expanded with the label of the edge running from the previous node to the current node.

As an example, the translation graph for the sentence 'Samstag und Februar sind gut, aber der vierte wäre besser' is shown in Figure 2. Actually, the graph is much bigger. In the figure, only those edges are shown which contributed to the construction of the best path.

3.1 Error Tolerant Match

To improve the coverage on unseen test data, it may be advantageous to allow for only approximative matching with the segments in the translation memory. The idea is to apply longer segments for syntactically better translations without losing too much as far as the content of the sentences is concerned. We use a weighted edit distance, i.e. each error (insertion, deletion, substitution) is associated with a score. Thereby, the deletion or insertion of typical filler words can be allowed, whereas the deletion or insertion of content words is avoided.

Hypotheses with to high a matching error score are discarded. A threshold proportional to the number of covered positions is used. Thus, longer translation patterns can be matched with more insertions, deletions and substitutions. A drawback of this is, however, that for long patterns mismatches on content words may occur.

Each transducer has its own list of insertion, deletion and substiution scores. Actually, only for those transducers where the translation patterns cover longer sequences of words and labels do we use error tolerant matching.

Error-tolerant matching may also help to compensate for speech recognition errors in the case of speech translations. In that case the confusion matrix obtained by comparing the recognizer output for the training speech data with the transliteration can be used.

3.2 Using a Language Model

The application of the transducers to a given source sentence yield a large number of target sentences. These are scored according to the cumulative scores of the applied translation patterns. As an independent and direct model of the likelihood of the target sentences a language model is applied. We use a word-based trigram language model (Sawaf et al., 2000). The logarithm of the language model probabilities is added to the transducer scores

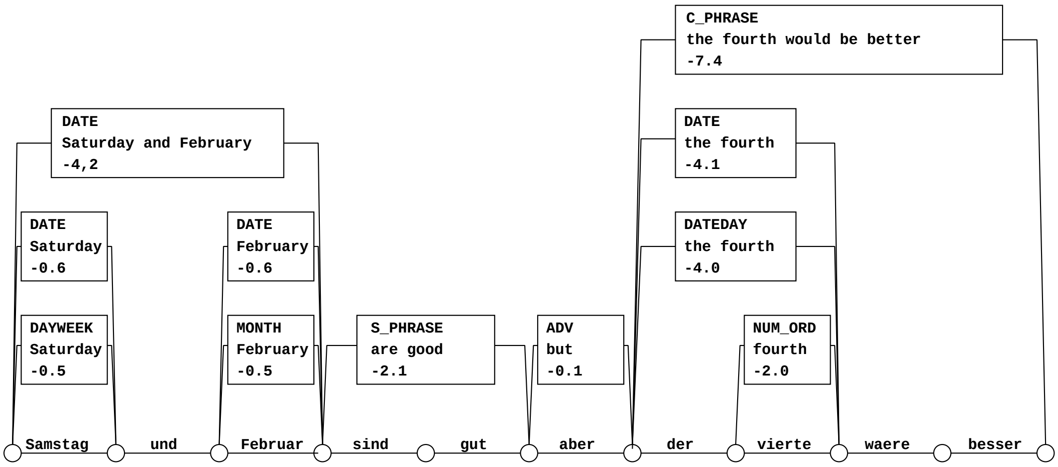

Figure 2: Translation example.

when the best path through the translation graph is extracted. A scaling factor allows for a bias on the effect of the language model.

4 Experiments and Results

In this section, we will give some results obtained with the cascaded transducer approach. Experiments were performed on the VERBMOBIL corpus. This corpus consists of spontaneously spoken dialogs in the appointment scheduling domain (Wahlster, 2000). A summary of the corpus used in the experiments is given in Table 3. In Table 4 the sizes of the special purpose transducers are given.

Table 3: Training and test conditions for the VERBMOBIL task. The trigram perplexity (PP) is given.

		German	English
Train	Sentences	34 465	
	Words	363 514	383 509
	Voc.	6 381	3 766
Test	Sentences	147	
	Words	1 968	2 173
	PP	–	19.7

The sentences from the training corpus were segmented into shorter segments using sentence marks as breakpoints. This resulted in 43 609 bilingual phrases running form 1 word up to 82 words in length. The longest

Table 4: Size of the transducers.

Transducer	Patterns
Names	442
Numbers	342
Spell	60
SimpleDate	161
CompoundDate	173
WordTags	6 714
Grammar	124

phrases were discarded as it is very unlikely that they will match other sentences. So, for the construction of the translation patterns only 40 000 sentence pairs were used, the longest sentences containing sixteen source words. Starting from those simple phrases, successively more transducers were applied up to the full cascade. A total of 15 682 translation patterns containing one or more labels resulted and nearly 4 500 sentence pairs became identical when words or word sequences were replaced by labels.

For a test corpus consisting of 147 sentences, the translations have been evaluated according to two measures (Nießen et al., 2000): Multi-reference word error rate (mWER): for each source sentence several good translations are given. The word error rate between the generated translation and the closest reference is calculated. Subjective sentence error rate (SSER): the translations are evaluated by a human examiner using a scale ranging from 0 to 10. The average of these values is linearly transformed to give the sentence error rate in percent.

4.1 Effect of Grammar

A simple translation memory without any categorization gives insufficient coverage on unseen test data. With the part-of-speech transducer we get one or more translations for each word in the vocabulary. But only by applying transducers which handle longer translation patterns is word reordering possible.

In Table 6 the results are given for different combinations of transducers. The baseline (T) is the combination of all special purpose transducers (name, spell, number, date, word tags) plus the simple translations patterns. Then the grammar was added and finally the compound translation patterns. The trigram language model for the target language was applied in selecting the best translation, but no error tolerant matching was allowed.

Table 6: Effect of bilingual grammar on translation quality: T = POS-tagging, G = grammar, C = compound translation patterns.

Transducer	mWER[%]	SSER[%]
T	41.2	25.8
TG	39.7	22.5
TGC	38.8	22.1

We observe a clear effect in word error rate and subjective sentence error rate. The use of the bilingual grammar, also very restricted, improves translation quality. Applying the compound translation patterns gives an additional small improvement.

In Table 5 a simple and a more involved example for the reordering effect of the bilingual grammar are given. The first translation pattern operates solely on the level of POS tags whereas the second example generates a hierarchical structure. We are not concerned whether the source sentence parses are correct, good translations is what we are looking for.

4.2 Effect of Language Model

The next experiment shows the effect of applying a language model for the target language. A word-based trigram language model was interpolated with the scores from the transducers. In Table 7 the effect of the scaling between the two models is shown.

There is a clear drop in the WER when switching on the language model. This is due to the fact, that several translation hypotheses have the same score from the transducers. So, it is rather by chance if the best translation for a given word is chosen. The language model for the target language helps in doing this.

Table 7: Effect of language model on word error rate and subjective sentence error rate.

LM Scale	mWER[%]	SSER[%]
0.0	49.3	31.8
0.2	38.8	23.5
0.5	38.8	22.1
1.0	39.4	23.8
5.0	42.6	27.4

There is a second benefit gained from the language model: sometimes the source sentence can be covered with only very short source patterns. That is to say, word context is hardly taken into account. With a language model context is brought into play again. If the language model scaling factor is increased too much translation quality deteriorates again. So, a good balance between both knowledge sources is necessary.

In Table 8 some examples which show the effect of the language model are given. The first translation is without language model, the second is the translation obtained when the language model score is added using a scaling factor of 0.5.

4.3 Effect of Error Tolerant Matching

Finally, the effect of error tolerant matching has been investigated. Only for the simple and compound translation patterns errors have been allowed in matching parts of the in-

Table 5: Example for the application of the bilingual grammar.

VP # PPER VMFIN PP VVINF # PPER VMFIN VVINF PP						
VP	{	PPER	{	ich # I # -0.1 }		
		VMFIN	{	m”ochte # want # -0.1 }		
		PP	{	APPR	{	mit # with # -0.1 }
			{	PPER	{	Ihnen # you # -0.1 }
				NP	{	ART { einen # a # 0.01 }
					{	NN { Termin # date # -0.1 }
				# a date # -2.09 }		
		# a date with you # -6.29 }				
		VVINF	{	vereinbaren # to arrange # -0.1 }		
# I want to arrange a date with you # -12.59 }						

Table 8: Examples for the effect of the language model.

	erst wieder ab dem sechzehnten.
no LM	starting from the sixteenth only again.
with LM	only starting from the sixteenth.

	ja, wunderbar. machen wir das so, und dann treffen wir uns dann in Hamburg.
no LM	yes, nice. will we do which right, after all we meet us after all in Hamburg.
with LM	fine. let us do it like that, and then we will meet then in Hamburg.

put sentences to stored translation patterns. The effect of increasing the error threshold is given in Table 9.

Table 9: Effect of error tolerant matching.

Errors per word	mWER[%]	SSER[%]
0.0	38.8	22.1
0.2	38.3	20.3
0.4	37.0	21.0
0.6	39.6	24.2

We see a considerable improvement when allowing for a small number of errors in matching the translation patterns to the input sentence. However, if the match gets too sloppy serious errors occur which alter the meaning of the sentence. For longer sequences of words the number of errors allowed becomes higher than the default score for substitutions. In such a case content words can be substituted.

An example of how the same source sentence gets different translations when more matching errors are allowed is given in Table 10.

5 Summary and future work

In this paper a translation approach based on cascaded finite state transducers has been presented. A small number of simple transducers is hand-crafted and then used to convert a bilingual corpus into a translation memory consisting of source pattern – target pattern pairs, which include category labels. Translation is then performed by applying the complete cascade of transducers.

With the simple heuristic for the translation scores a language model for the target language is paramount to select good translations. Error-tolerant matching improves translation quality.

Experiments have shown the potential of this approach for machine translation. Good coverage on unseen test data could be obtained. A major advantage of this translation

Table 10: Examples for the effect of error tolerant matching.

	ja , wunderbar . machen wir das so , und dann treffen wir uns dann in Hamburg .
0.0	fine . let us do it like that , and then we will meet then in Hamburg .
0.2	fine . let us do that , and then we will meet in Hamburg .
0.4	fine . let us do it like that , and then we will meet in Hamburg .
0.6	fine . let us do it like that , and then we will meet in your office .

method is that it breaks the middle ground between direct translation methods like simple translation memory or word-based statistical translation and transfer based methods involving deep linguistic analysis of the input. In fact, the cascaded transducer approach allows for building quickly a first version and improving translation quality by gradually adding more linguistic and domain specific knowledge.

We expect further improvement by assigning translation scores according to corpus statistics. This will be the main focus for future work.

Acknowledgement. This work was partly supported by the German Federal Ministry of Education, Science, Research and Technology under the Contract Number 01 IV 701 T4 (Verbmobil).

References

S. Abney. 1997. Part-of-speech tagging and partial parsing. In S. Young and G. Bloothooft, (Eds.), *Corpus-based Methods in Language and Speech Processing*, pages 118–136. Kluwer Academic Publishers, Dordrecht, Boston, London.

J. C. Amengual, J. M. Benedi, F. Casacuberta, A. Castano, A. Castellanos, V. M. Jimenez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J. M. Vilar. 2000. The Eutrans-I speech translation system. *Machine Translation, Special Issue, forthcoming.*

R. D. Brown. 1996. Example-based machine translation in the pangloss system. In *Proc. of COLING '96: The 16th Int. Conf. on Computational Linguistics*, pages 169–174, Copenhagen, Denmark, August.

H. Kitano. 1993. A comprehensive and practical model of memory-based machine translation. In R. Bajcsy, editor, *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence*, pages 1276–1282. Morgan Kaufmann.

S. Nießen, F. J. Och, G. Leusch, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *2nd Int. Conf. on Language Resources and Evaluation*, pages 39–45, Athens, Greece, May.

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, USA, June.

H. Sawaf, K. Schütz, and H. Ney. 2000. On the use of grammar based language models for statistical machine translation. In *6th Int. Workshop on Parsing Technologies*, pages 231–241, Trento, Italy, February.

A. Schiller, S. Teufel, and C. Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart and Universität Tübingen. `http://www.sfs.npphil.uni-tuebingen.de/Elwis/stts/stts.html`.

S. Vogel and H. Ney. 2000. Construction of a hierarchical translation memory. In *Proc. of COLING 2000: The 18th Int. Conf. on Computational Linguistics*, pages 1131–1135, Saarbrücken, Germany, July.

Wahlster, W. (Ed.) 2000. *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer-Verlag Heidelberg.

Y.-Y. Wang and A. Waibel. 1997. Decoding algorithm in statistical translation. In *Proc. of the 35th Annual Conf. of the Association for Computational Linguistics*, pages 366–372, Madrid, Spain, July.

D. Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In *Proc. of the 34th Annual Conf. of the Association for Computational Linguistics*, Santa Cruz, CA, pages 152–158, June.