ON THE USE OF GRAMMAR BASED LANGUAGE MODELS FOR STATISTICAL MACHINE TRANSLATION

Hassan Sawaf, Kai Schütz, Hermann Ney
Computer Science Department – Lehrstuhl für Informatik VI
RWTH Aachen – University of Technology
D-52056 Aachen, Germany
sawaf@cs.rwth-aachen.de

Abstract

In this paper, we describe some concepts of language models beyond the usually used standard trigram and use such language models for statistical machine translation.

In statistical machine translation the language model is the a-priori knowledge source of the system about the target language. One important requirement for the language model is the correct word order, given a certain choice of words, and to score the translations generated by the translation model $\Pr(f_1^J|e_1^I)$, in view of the syntactic context.

In addition to standard m-grams with long histories, we examine the use of Part-of-Speech based models as well as linguistically motivated grammars with stochastic parsing as a special type of language model. Translation results are given on the VERBMOBIL task, where translation is performed from German to English, with vocabulary sizes of 6500 and 4000 words, respectively.

1 Introduction to Statistical Machine Translation

In this paper, we describe some methods of how to use more structured information in language modelling to improve statistical machine translation (SMT). The organisation of the paper is as follows: In this section a short introduction to SMT is given. The following section gives an overview of the different language model approaches to SMT, then our definition of performance measures and experimental results follow.

The goal of machine translation is the translation of a text given in some source language into a text in the target language. We are given a source string $f_1^J = f_1 \ldots f_j \ldots f_J$, which is to be translated into a target string $e_1^I = e_1 \ldots e_i \ldots e_I$. Among all possible target strings, we will choose the string with the highest probability:

$$\begin{aligned} \hat{e}_1^I &= \arg\max_{e_1^I} \{\Pr(e_1^I|f_1^J)\} \\ &= \arg\max_{e_1^I} \{\Pr(e_1^I) \cdot \Pr(f_1^J|e_1^I)\} \quad . \qquad (1) \end{aligned}$$

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. $\Pr(e_1^I)$ is the *language model* (LM) of the target language, which will be investigated in this paper, whereas $\Pr(f_1^J|e_1^I)$ is the *translation model* that is the main topic in [1, 9, 11, 12, 13, 14]. In this work the alignment template approach as described in [11] is used.

2 Language Models for SMT

Especially for the task of translation, the need of more restriction in the LM prove to be necessary. Restriction of the LM of the target language intuitively should improve translation quality in that way that the translation model should allow many alignment possibilities that are then restricted by the language model. Some heuristics help SMT systems to perform better, for example by re-ordering the source sentence as in [13] or by producing permutations in the search that are scored by a LM as in [11, 17, 18]. The advantage of the latter approach is that the reordering is integrated into the search and needs not be done as a preprocessing step. For this approach a more restricting LM, in terms of linguistic constraints and the ability to model long range dependencies is helpful.

In the work of [1], where the translation direction is French to English, the robust m-gram models have been used, as they cover the need for the analysed task well. For French-English the word order difference is mostly of a local nature, LMs that model embedded long range structures seem not to be necessary.

In this paper we show experiments using following types of LMs: word based m-gram models, class based m-gram models and context free grammar based models.

2.1 Standard Word based m-Grams

The m-gram LMs have their strength in their simplicity and reliability in robustness. That is the reason why they are widely used in automatic speech recognition. They are derived as follows:

$$\begin{aligned} \Pr(e_1^I) &= \prod_{i=1}^{I} \Pr(e_i|e_1^{i-1}) = \prod_{i=1}^{I} \Pr(e_i|h_i) \\ &= \prod_{i=1}^{I} p(e_i|e_{i-m+1}^{i-1}) \quad . \qquad (2) \end{aligned}$$

In equation (2) an m-gram LM is used. For the experiments shown in this paper we used absolute discounting with interpolation, since this method has succeeded in outperforming the backing-off strategy and linear interpolation [7]. On the other hand the implementation of absolute discounting with interpolation has a low cost regarding calculation complexity [16].

2.2 Part-Of-Speech based m-Grams

To cope with the problem of sparse data, class based LMs have been studied. The classes can be extracted automatically using clustering algorithms as in [6, 8] or they are defined by linguistic experts. Here we want to show the use of the linguistically motivated Part-of-Speech (POS) based m-gram LMs.

The basis of class based m-gram LMs can be written as follows:

$$\begin{aligned} \Pr(e_1^I) &= \sum_{c_1^I} \Pr(e_1^I, c_1^I) = \sum_{c_1^I} \Pr(e_1^I|c_1^I) \Pr(c_1^I) \\ &= \sum_{c_1^I} \prod_{i=1}^{I} p(e_i|c_i, h_i) p(c_i|h_i) \qquad \text{with } h_i = c_1^{i-1} \\ &\cong \prod_{i=1}^{I} \max_{c_i} \{p(e_i|c_i, h_i) \cdot p(c_i|h_i)\} \quad . \qquad (3) \end{aligned}$$

In equation (3) the LM is divided into two submodels: the classification model $p(e_i|c_i,h_i)$ and the class sequence model $p(c_i|h_i)$, where e_i denotes the ith word of the hypothesised sentence, c_i a classification of this word and h_i is the sequence of preceding word classes $h_i = c_1^{i-1}$ or, using an m-gram, $h_i = c_{i-m+1}^{i-1}$.

As shown in formula (3) we use the simplification that we perform the maximisation *before* calculating the probability of the whole word sequence. This means that the found maximum is only a local maximum. Instead the maximisation should have taken place after the calculation for the whole sequence, which would require a search similar to the monotone search [12].

Using non-unique class membership adds the problem of smoothing the class probability $p(e_i|c_i,h_i)$ in formula (3). It shows that absolute discounting for disambiguation of the class here performs best. Formally the absolute discounting of the class probability is as follows:

$$p(e_i|c_i,h_i) = \max\left\{\frac{N(e_i,c_i,h_i)-b_c}{N(c_i,h_i)},0\right\} + b_c \cdot \frac{W-n_0(c_i,h_i)}{N(c_i,h_i)} \cdot \beta(e_i|c_i,\overline{h_i}) \quad , \tag{4}$$

where $N(.)$ denotes the count of an event. W is the size of the vocabulary and $n_0(c_i,h_i)$ denotes the number of words not seen with the class c_i and class history h_i. $\beta(e_i|c_i,\overline{h_i})$ describes a less specific distribution with the generalised class history $\overline{h_i}$. The probability distributions for the class probabilities used in the present study have the same order as the class sequence probabilities, for example if the class sequence model is a trigram, the class probability is

$$\begin{aligned}
p(e_i|c_i,c_{i-1},c_{i-2}) &= \max\left\{\frac{N(e_i,c_i,c_{i-1},c_{i-2})-b_{c,3}}{N(c_i,c_{i-1},c_{i-2})},0\right\} \qquad (5)\\
&\quad + b_{c,3} \cdot \frac{W-n_0(c_i,c_{i-1},c_{i-2})}{N(c_i,c_{i-1},c_{i-2})} \cdot p(e_i|c_i,c_{i-1}) \quad ;\\
p(e_i|c_i,c_{i-1}) &= \max\left\{\frac{N(e_i,c_i,c_{i-1})-b_{c,2}}{N(c_i,c_{i-1})},0\right\} \qquad (6)\\
&\quad + b_{c,2} \cdot \frac{W-n_0(c_i,c_{i-1})}{N(c_i,c_{i-1})} \cdot p(e_i|c_i) \quad ;\\
p(e_i|c_i) &= \max\left\{\frac{N(e_i,c_i)-b_{c,1}}{N(c_i)},0\right\} \qquad (7)\\
&\quad + b_{c,1} \cdot \frac{W-n_0(c_i)}{N(c_i)} \cdot p_{c,0} \quad ;\\
p_{c,0} &= \frac{1}{W} \quad . \qquad (8)
\end{aligned}$$

As with absolute discounting in standard language modelling the calculation of the probability $p(e_i|c_i,c_{i-1},c_{i-2})$ consists of a *trigram* (5), a *bigram* (6), a *unigram* (7) and a *zerogram* (8) portion, which is given by the inverse of the number of vocabulary entries.

2.3 Stochastic Context Free Parser

When using m-gram LMs, some linguistic phenomena are not captured very well, because they do not model long range dependencies and embedded structures. A possible solution to this problem the use of context free grammars (CFG) as LMs [2, 18].

Formally a CFG is a quadruple $\mathcal{G} = (V_N, V_T, R, S)$, where V_N is the set of all nonterminals, V_T the set of terminals, R is the set of rules and S denominates the starting symbol that has to cover the analysed sentence.

To use CFG we implemented a stochastic parser using the stochastic version of the algorithm introduced by Cocke, Younger and Kasami [4, 19], by assigning a probability $p(r: A_n \rightarrow A_\alpha A_\beta | A_n)$ to each rule $r : A_n \rightarrow A_\alpha A_\beta$. The best hypothesis is the sequence of rule applications that produces the whole sentence from the nonterminal S and has the highest probability. The probabilities are trained using the Viterbi approximation so that only the best parse is used to calculate the rule probabilities.

If we assume the rule probabilities to be independent of each other and the class probabilities of the Part-of-Speech tags distributed uniformly, the LM probabilities can then be written as:

$$\begin{aligned}
\Pr(e_1^I) &= \sum_{r_1^N} p(r_1^N : S \xrightarrow{N} e_1^I | S) \\
&= \sum_{r_1^N} \prod_{n=1}^{N} p(r_n | A_n) \quad , \quad \text{with } A_n \text{ being the left-hand side of rule } r_n \quad , \\
&\cong \max_{r_1^N} \prod_{n=1}^{N} p(r_n | A_n) \quad . \qquad (9)
\end{aligned}$$

In order to include the tagging process into the stochastic parser, we change the format of the standard Chomsky Normal form (CNF) in the following way. We distinguish three types of rules and associated probabilities:

- **tagging rule**

  $c_i \rightarrow e_i$

  with probability $p(c_i \rightarrow e_i | c_i)$,

- the so-called **lexical rule**

  $A_\alpha \rightarrow c_i$

  with probability $p(A_\alpha \rightarrow c_i | A_\alpha)$, and

- the so-called **structure rule**

  $A_\alpha \rightarrow A_\beta A_\gamma$

  with probability $p(A_\alpha \rightarrow A_\beta A_\gamma | A_\alpha)$.

For words that are not observed in the training corpus, a simple backing-off smoothing technique is used for tagging to be able to tag these unknown words. Note that, according to equation (9), the optimal POS tags are determined only *after* the whole sentence has been parsed. In that sense the tagger uses a global sentence level criterion rather than a local decision criterion.

The search for the best parse is done using dynamic programming over the positions $1 < i, j < I$. The recursion formula for the stochastic Cocke-Younger-Kasami-style parser (SCYK) using the Viterbi approximation is based on partial hypotheses Q derived from (9) with

$$Q(j, i | A_n \rightarrow A_\alpha A_\beta) = p(A_n \rightarrow A_\alpha A_\beta | A_n) \max_{j<l<i-1} \big(Q(j, l | A_\alpha) \; Q(l+1, i | A_\beta) \big) \quad .$$

We use two speed-up methods for the SCYK:

top-down filtering: The parsing proceeds mainly bottom up, apart from a top-down filtering method that does not affect the parsing accuracy. The filtering limits the number of nonterminals that can produce the hypothesised sentence fragment, allowing only nonterminals that can be reached from a special position within the sentence.

bounding: Another implemented feature to improve speed is that a calculation is cancelled, if the scores of partial hypotheses do not reach a lower bound. The main idea is that a product of two probabilities cannot be higher than the smaller of the two probabilities.

To be able to use grammars that are not in CNF, we implemented an algorithm that converts any CFG into CNF. The standard algorithm is sketched in table 1 [3].

Table 1: Standard algorithm for converting a general CFG into CNF.

```
while $\exists$ rule $r : A \rightarrow A'$ in productions $R$
    foreach occurrence of nonterminal $A$ in $R$
        foreach rule $r'$ with $A$ on the left-hand side
            add rules by applying $r'$ to all rules with $A$ on the
            right hand side into $R$
    remove rule $r$ from $R$
while $\exists$ rule $r : A \rightarrow A_1 A_2 A_3 \ldots A_I$ in $R$ which violates CNF
    add new nonterminal $A^*$ to $V_N$
    add new rule $r^*$ with $r^* : A^* \rightarrow A_1 A_2$ to $R$
    rewrite rule $r$ as $r : A \rightarrow A^* A_3 \ldots A_I$
```

The problem of this algorithm is that the number of newly added rules increases rapidly so that the memory requirement for the CNF grammar is very high and also time complexity increases. The reason of this increase is that the complexity of the SCYK is $O(|V_N| \cdot |R| \cdot I^3)$. According to this formula we would prefer a CNF grammar with minimal numbers $|R|$ of rules and $|V_N|$ nonterminals.

Here, we introduce an algorithm that uses bigram frequencies to determine which pair of nonterminals to choose when generating a new rule (see algorithm in table 2). Thus we can reduce the number of rules significantly.

Table 2: Improved algorithm for converting a general CFG into CNF.

```
while $\exists$ rule $r : A \rightarrow A'$ in productions $R$
    foreach occurrence of nonterminal $A$ in $R$
        foreach $r'$ with $A$ on the left-hand side
            add rules by applying $r'$ to all rules with $A$ on
            the right hand side into $R$
    remove rule $r$ from $R$
while $\exists$ rule $r : A \rightarrow A_1 A_2 \ldots A_I$ in $R$ which violates CNF
    create 2-dimensional bigram count table $b(\cdot,\cdot)$ over
    all nonterminals $A_\alpha, A_\beta$ on the right hand side
    add new nonterminal $A^*$ to $V_N$
    add new rule $r^*$ with $r^* : A^* \rightarrow A_\alpha A_\beta$ to $R$,
    where $b(A_\alpha, A_\beta)$ has highest value
    rewrite all rules, replacing successive nonterminals
    $A_\alpha A_\beta$ on the right hand side
```

There must be some considerations as to what to do regarding the rule probabilities when transforming the grammar. The easiest way to handle the probabilities is as follows: when applying a rule, the rule probabilities are multiplied as usual. When adding a rule to the productions R, the new rule r' has probability $p(r' : A' \rightarrow A_\alpha A_\beta | A') = 1$.

The implemented parser is constructed in such a way so that the hypothesised sentence is processed left-to-right that means the chart for the SCYK is constructed along the covered positions of the sentence instead of the standard way, where the chart is constructed along the depth of the parse subtrees. In such a way, we can use the parser as a left-to-right LM that can be embedded into the translation approaches, e.g. described in [9, 11, 12] or in speech recognition systems that build up the recognised sentence incrementally.

2.4 Linear Interpolation of Language Models

Experiments show that every LM type has some advantages compared to the other LMs, but also bears some weakness so that it would be best to use all LMs at the same time. An m-gram LM for instance has its strength in robustness and we expect the grammar based LM to model long range dependencies better.

A very easy method to combine two LMs $p_1(\cdot)$ and $p_2(\cdot)$ is to use *linear interpolation* [8] at the full-sentence level:

$$Pr(e_1^I) = (1-\alpha) \cdot p_1(e_1^I) + \alpha \cdot p_2(e_1^I) \quad \text{with } 0 < \alpha < 1 \quad .$$

3 Parsing Performance

The "VERBMOBIL Task" [15] is a speech translation task in the domain of appointment scheduling, travel planning and hotel reservation. The translation direction is from German to English which poses special problems due to the big difference in the word order of the two languages.

To perform experiments with the SCYK parser, we used the English part of the VERBMOBIL treebank [5]. Table 3 shows some statistics about the investigated corpora. For the performance tests of the parser, we used the standard training and test set that consist of about 9000 and 500 trees, respectively, where every tree corresponds to one sentence. To train the parser for rescoring the translation results, we used all available trees in the VERBMOBIL treebank, i.e. about 21,000 sentences.

Table 3: VERBMOBIL treebank characteristics.

corpus	train	test	all
trees	9126	500	20,889
running words	81,382	4331	185,084
vocabulary size	1710	498	2168
avg. sentence length	8.88	8.66	8.86
avg. tree depth	6.58	6.55	6.58
avg. nodes per tree	11.25	10.8	11.12
$\|V_N\|$	96		99
$\|V_T\|$	72		76
$\|R\|$	4287		6260
avg. rule length	3.54		3.68
unseen words		31	
unseen rules		143	
trees with unseen rules		113	

Results for the time complexity are shown in table 4. As described in chapter 2.3 we used two methods, namely the top-down filtering and bounding to speed up the parsing process. These two methods are included in 4.

Table 4: Time performance of SCYK of the speed-up methods.

	avg. time/sentence
baseline	246.7 ms
+ top-down filtering	226.4 ms
+ bounding	63.78 ms
+ bounding + top-down filtering	62.82 ms

The enhanced transformation method for converting a generic grammar into Chomsky Normal form compared to the standard method is presented in table 5. The corpus used here is the above mentioned VERBMOBIL standard treebank test corpus.

Table 5: Results for conversion into Chomsky Normal form.

method	$\|V_N\|$	$\|R\|$
no conversion	99	6260
standard	10932	17017
improved	1170	7255

Table 6 shows the parsing performance on the VERBMOBIL treebank test corpus. When looking at the results shown in table 6, we should keep in mind that there was a certain number of unseen words (see table 3). Due to this number of unseen words the tagging accuracy decreases from 96.73% to 95.87%, the complete match from 51.6% to 49.6% and bracketing recall and precision drop by about one percent absolute respectively.

Table 6: Parsing results on standard testset of VERBMOBIL.

	SCYK
number of sentences	500
number of unparsed sentences	0
number of valid sentences	500
bracketing recall	84.84 %
bracketing precision	84.82 %
complete match	49.60 %
average crossing	0.43
no crossing	80.40 %
2 or less crossing	93.80 %
tagging accuracy	95.87 %

When using no smoothing technique for tagging, the parser would not parse 31 out of the 500 sentences. It has also to be mentioned that 22.6% of the test sentences contain rules that cannot be matched by the rules generated from the training corpus. That means, the highest possible value for complete match would be 77.4%.

4 Translation Results

4.1 The VERBMOBIL Task

For translation experiments we used the bilingual text corpus of the VERBMOBIL task. The text input was obtained by manually transcribing the spontaneously spoken sentences. There was no constraint on the length of the sentences, and some of the sentences in the test corpus contain more than 50 words. Therefore, for text input, each sentence was split into shorter units using the punctuation marks. The segments thus obtained were translated separately, and the final translation was obtained by concatenation.

Table 7 shows a summary of the corpus used for the experiments. Here the term word refers to full-form word as there is no morphological processing involved.

Table 7: Training and test conditions for the VERBMOBIL Task.

		German	English
Train	sentences	34 465	
	running words	363 514	383 509
	vocabulary size	6 381	3 766
Test	sentences	147	
	running words	1 968	2 173

4.2 Performance Measures

In the translation experiments, we use the following performance measures [10]:

- mWER (multi-reference word error rate):
  A known weakness of the *word error rate* that is widely used for speech recognition is that a translation of a given sentence is not unique so that there can be more than one translation for a source sentence. A possible solution for this is to use a set of several possible translations for each source sentence. The mWER is the Levenshtein distance to the most similar sentence of this set.

- SSER (subjective sentence error rate):
  For a more detailed analysis, subjective judgements by test persons are necessary. Each translated sentence is judged by a human examiner according to an error scale from 0.0 to 1.0 in eleven steps. A rating of 0.0 means that the translation is semantically and syntactically correct and a rating of 1.0 means that the sentence is semantically wrong, i.e. either the produced sentence has no sense at all or the produced sense does not convey the sense of the source sentence. The human examiner was offered the translated sentences for the different LMs at the same time.

4.3 Translation Results for VERBMOBIL

In Table 8 some results for the different LMs for SMT are presented. The results are calculated by rescoring the 100 best hypotheses of each sentence of the test set using the alignment template approach. The hypotheses are then re-calculated by multiplying the translation model scores and the language model scores like in formula 1. The pure POS-based LMs for SMT show relatively poor performance compared to word based LMs, only if using an interpolation of both word and POS-based LMs the performance improves slightly. Table 8 contains the results of using the different LMs in terms of mWER and SSER.

Table 8: Translation performance on VERBMOBIL

LM		mWER[%]	SSER[%]
word based	2-gram	37.76	23.54
	3-gram	35.40	22.59
	4-gram	35.61	23.20
	5-gram	35.45	22.45
	10-gram	34.89	22.11
POS based	2-gram	39.00	24.83
	3-gram	38.34	22.52
	4-gram	38.09	23.47
	5-gram	37.36	23.95
	10-gram	37.43	24.15
stochastic CFG		36.55	22.31
linear interpolation of			
word 5-gram + POS 5-gram		34.13	21.22
POS 5-gram + SCFG		31.95	20.48

Using long history length seems to perform better for m-gram LMs both word and POS-based. The SCFG alone does not improve translation results compared to the word based LM. The interpolation of both POS-based LM and SCFG achieves the best results. Linear interpolation of word and POS-based LMs also achieves better results than POS-based or word based LMs alone but does not reach the performance of the combination of POS-based LM and SCFG.

Analysing the sentences chosen from the different LM types we can observe that the SCFG is superior to the m-gram LMs in modelling nested structures and long range dependencies as can be seen in table 9. In each of the two cases for linear interpolation shown in table 9, the interpolation factors were 0.5, which produced the best results.

The sentences show that the syntactic quality increases when using more linguistic information for SMT. The third sentence in table 9 shows the advantage of using grammar based LMs. The corresponding sentence of the word based model contains the problem that the verb group for this sentence in the source language is composed of two parts: *am* produced from the first part *habe* and *make* produced from *ausgemacht*, are positioned relatively far from each other within the sentence. The coherence of these two words is thus not detected by the m-gram LMs. The SCFG, however, can detect this and constructs a better verb phrase.

For the third translation example in table 9 it should be noted that the list of 100 sentences does not include the correct sentence. After adding the correct sentence to the list, the system produced the correct translation.

5 Conclusion

In this paper we discussed the use of linguistically motivated language models for statistical machine translation, namely Part-of-Speech based m-gram models and stochastic context free grammars. The results of the different language model types and the interpolation of the language models are then

Table 9: Translation examples on VERBMOBIL task and reference translation.

German	also, ich habe Unterlagen über Flüge da.
reference	well, I have information about flights here.
word based	well, I have about brochures flights.
word+POS	well, I have flights about brochures.
POS+SCFG	well, I have papers about flights then.
German	ich könnte erst eigentlich jetzt wieder dann November vorschlagen.
reference	actually I could only suggest November then.
word based	now again then actually I could only suggest November.
word+POS	I could only suggest again now actually of November then.
POS+SCFG	I could suggest again then now actually first of November.
German	also ausgerechnet habe ich am dritten Juli schon einen Zahnarzttermin ausgemacht.
reference	well, on the third of July I have already made a dentist's appointment.
word based	so, I am on the third of July already make a dentist appointment.
word+POS	well, I have the third of July already make a dentist appointment.
POS+SCFG	well, on the third of July I have got already make a dentist appointment.

presented on the VERBMOBIL task. It shows to be a promising approach to use a stochastic context free parser as syntax-restricting language model and to interpolate it with a Part-of-Speech based language model.

In the near future the language models introduced here will be integrated into the translation process itself instead of using rescoring. In [7] different interpolation methods are compared and the linear interpolation was found to be worse than log-linear interpolation methods. Therefore these interpolation methods should be examined for language models for statistical machine translation in the future. Also refined grammar based models should be investigated, for instance usage of lexicalized grammars or stochastic attribute grammars. The usage of morphological analysis should achieve a gain in syntactic quality for the produced sentences.

Acknowledgement

This work has been partly supported as part of the VERBMOBIL project (contract number 01 IV 601 A) by the German Federal Ministry of Education, Science, Research and Technology.

The authors would like to thank Franz Josef Och for his fruitful discussions and Richard Zens for his support in performing the tests with the m-gram LMs.

References

[1] P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer: The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263-311, 1993.

[2] C. Chelba, F. Jelinek: Recognition Performance of a Structured Language Model. *6th European Conference on Speech Communication and Technology*, pp. 1567-1570, Budapest, Hungary, September 1999.

[3] J. E. Hopcroft, J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, Reading, MA, 1979.

[4] T. Kasami: An Efficient Recognition and Syntax Analysis Algorithm for Context Free Languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA, 1965.

[5] V. Kordoni: Stylebook for the English Treebank in VERBMOBIL. Technical report, University of Tübingen, Tübingen, Germany, 1998.

[6] S. C. Martin, J. Liermann, H. Ney: Algorithms for Bigram and Trigram Word Clustering. *Speech Communication*, Vol. 24, No. 1, pp. 19-37, April 1998.

[7] S. C. Martin, C. Hamacher, J. Liermann, H. Ney: Assessment of Smoothing Methods and Complex Stochastic Language Modelling. *6th European Conference on Speech Communication and Technology*, pp. 1939-1942, Budapest, Hungary, September 1999.

[8] H. Ney, F. Wessel, S. C. Martin: Statistical Language Modelling Using Leaving-One-Out. In S. Young, G. Bloothooft (eds.): *Corpus-Based Methods in Speech and Language*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 174-207, 1997.

[9] S. Nießen, S. Vogel, H. Ney, C. Tillmann: A DP-Based Search Algorithm for Statistical Machine Translation. *36th Annual Conference of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 960-967, Montreal, Canada, August 1998.

[10] S. Nießen, F. J. Och, G. Leusch: An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. *submitted to 2nd International Conference on Language Resources and Evaluation*, Athens, Greece, May 2000.

[11] F. J. Och, C. Tillmann, and H. Ney: Improved Alignment Models for Statistical Machine Translation. *Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20-28. University of Maryland, College Park, MD, June 1999.

[12] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga: A DP-based Search Using Monotone Alignments in Statistical Translation. *35th Annual Conference of the Association for Computational Linguistics*, pp. 289-296, Madrid, Spain, July 1997.

[13] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, H. Sawaf: Accelerated DP Based Search for Statistical Translation. *5th European Conference on Speech Communication and Technology*, pp. 2667-2670, Rhodos, Greece, September 1997.

[14] S. Vogel, H. Ney, C. Tillmann: HMM-Based Word Alignment in Statistical Translation. *International Conference on Computational Linguistics*, pp. 836-841, Copenhagen, Denmark, August 1996.

[15] W. Wahlster: VERBMOBIL: Translation of Face-to-Face Dialogs. *Machine Translation Summit IV*, pp. 127-135, Kobe, Japan, 1993.

[16] F. Wessel, S. Ortmanns, H. Ney: Implementation of Word Based Statistical Language Models. *2nd SQEL Workshop on Multi-Lingual Information Retrieval Dialogues*, pp. 55-59, Pilsen, Czech Republic, April 1997.

[17] D. Wu: A Polynomial-Time Algorithm for Statistical Machine Translation. *34rd Annual Conference of the Association for Computational Linguistics*, pp. 152-158, Santa Cruz, CA, 1996.

[18] D. Wu, H. Wong: Machine Translation with a Stochastic Grammatical Channel. *36th Annual Conference of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 1408-1415, Montreal, Canada, August 1998.

[19] D. H. Younger: Recognition and Parsing of Context Free Languages in Time n^3. *Information and Control*, 10:2, pp. 189-208, 1967.