Reranking Translation Hypotheses Using Structural Properties

Saša Hasan, Oliver Bender, Hermann Ney
Chair of Computer Science VI
RWTH Aachen University
D-52056 Aachen, Germany
{hasan,bender,ney}@cs.rwth-aachen.de

Abstract

We investigate methods that add syntactically motivated features to a statistical machine translation system in a reranking framework. The goal is to analyze whether shallow parsing techniques help in identifying ungrammatical hypotheses. We show that improvements are possible by utilizing supertagging, lightweight dependency analysis, a link grammar parser and a maximum-entropy based chunk parser. Adding features to n-best lists and discriminatively training the system on a development set increases the BLEU score up to 0.7% on the test set.

1 Introduction

Statistically driven machine translation systems are currently the dominant type of system in the MT community. Though much better than traditional rule-based approaches, these systems still make a lot of errors that seem, at least from a human point of view, illogical.

The main purpose of this paper is to investigate a means of identifying ungrammatical hypotheses from the output of a machine translation system by using grammatical knowledge that expresses syntactic dependencies of words or word groups. We introduce several methods that try to establish this kind of linkage between the words of a hypothesis and, thus, determine its well-formedness, or "fluency". We perform rescoring experiments that rerank n-best lists according to the presented framework.

As methodologies deriving well-formedness of a sentence we use supertagging (Bangalore and Joshi, 1999) with lightweight dependency analysis (LDA)[1] (Bangalore, 2000), link grammars (Sleator and Temperley, 1993) and a maximum-entropy (ME) based chunk parser (Bender et al., 2003). The former two approaches explicitly model the syntactic dependencies between words. Each hypothesis that contains irregularities, such as broken linkages or non-satisfied dependencies, should be penalized or rejected accordingly. For the ME chunker, the idea is to train n-gram models on the chunk or POS sequences and directly use the log-probability as feature score.

In general, these concepts and the underlying programs should be robust and fast in order to be able to cope with large amounts of data (as it is the case for n-best lists). The experiments presented show a small though consistent improvement in terms of automatic evaluation measures chosen for evaluation. BLEU score improvements, for instance, lie in the range from 0.3 to 0.7% on the test set.

In the following, Section 2 gives an overview on related work in this domain. In Section 3 we review our general approach to statistical machine translation (SMT) and introduce the main methodologies used for deriving syntactic dependencies on words or word groups, namely supertagging/LDA, link grammars and ME chunking. The corpora and the experiments are discussed in Section 4. The paper is concluded in Section 5.

2 Related work

In (Och et al., 2004), the effects of integrating syntactic structure into a state-of-the-art statistical machine translation system are investigated. The approach is similar to the approach presented here:

[1] In the context of this work, the term LDA is not to be confused with *linear discriminant analysis*.

firstly, a word graph is generated using the baseline SMT system and n-best lists are extracted accordingly, then additional feature functions representing syntactic knowledge are added and the corresponding scaling factors are trained discriminatively on a development n-best list.

Och and colleagues investigated a large amount of different feature functions. The field of application varies from simple syntactic features, such as IBM model 1 score, over shallow parsing techniques to more complex methods using grammars and intricate parsing procedures. The results were rather disappointing. Only one of the simplest models, i.e. the implicit syntactic feature derived from IBM model 1 score, yielded consistent and significant improvements. All other methods had only a very small effect on the overall performance.

3 Framework

In the following sections, the theoretical framework of statistical machine translation using a direct approach is reviewed. We introduce the supertagging and lightweight dependency analysis approach, link grammars and maximum-entropy based chunking technique.

3.1 Direct approach to SMT

In statistical machine translation, the best translation $\hat{e}_1^{\hat{I}} = \hat{e}_1 \ldots \hat{e}_i \ldots \hat{e}_{\hat{I}}$ of source words $f_1^J = f_1 \ldots f_j \ldots f_J$ is obtained by maximizing the conditional probability

$$\begin{aligned} \hat{e}_1^{\hat{I}} &= \underset{I,e_1^I}{\operatorname{argmax}} \{Pr(e_1^I|f_1^J)\} \\ &= \underset{I,e_1^I}{\operatorname{argmax}} \{Pr(f_1^J|e_1^I) \cdot Pr(e_1^I)\} \end{aligned} \tag{1}$$

using Bayes decision rule. The first probability on the right-hand side of the equation denotes the translation model whereas the second is the target language model.

An alternative to this classical source-channel approach is the direct modeling of the posterior probability $Pr(e_1^I|f_1^J)$ which is utilized here. Using a log-linear model (Och and Ney, 2002), we obtain

$$Pr(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{{e'}_1^{I'}} \exp\left(\sum_{m=1}^M \lambda_m h_m({e'}_1^{I'}, f_1^J)\right)}, \tag{2}$$

where λ_m are the scaling factors of the models denoted by feature functions $h_m(\cdot)$. The denominator represents a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it during the search process, leading to the following decision rule:

$$\hat{e}_1^{\hat{I}} \;=\; \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{3}$$

This approach is a generalization of the source-channel approach. It has the advantage that additional models $h(\cdot)$ can be easily integrated into the overall system. The model scaling factors λ_1^M are trained according to the maximum entropy principle, e.g., using the GIS algorithm. Alternatively, one can train them with respect to the final translation quality measured by an error criterion (Och, 2003). For the results reported in this paper, we optimized the scaling factors with respect to a linear interpolation of word error rate (WER), position-independent word error rate (PER), BLEU and NIST score using the Downhill Simplex algorithm (Press et al., 2002).

3.2 Supertagging/LDA

Supertagging (Bangalore and Joshi, 1999) uses the Lexicalized Tree Adjoining Grammar formalism (LTAG) (XTAG Research Group, 2001). Tree Adjoining Grammars incorporate a tree-rewriting formalism using elementary trees that can be combined by two operations, namely substitution and adjunction, to derive more complex tree structures of the sentence considered. Lexicalization allows us to associate each elementary tree with a lexical item called the *anchor*. In LTAGs, every elementary tree has such a lexical anchor, also called head word. It is possible that there is more than one elementary structure associated with a lexical item, as e.g. for the case of verbs with different subcategorization frames.

The elementary structures, called initial and auxiliary trees, hold all dependent elements within the same structure, thus imposing constraints on the lexical anchors in a local context. Basically, supertagging is very similar to part-of-speech tagging. Instead of POS tags, richer descriptions, namely the elementary structures of LTAGs, are annotated to the words of a sentence. For this purpose, they are called *supertags* in order to distinguish them from ordinary POS tags. The result is an “almost parse” because of the dependencies

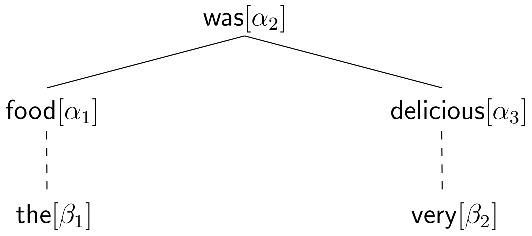

Figure 1: LDA: example of a derivation tree, β nodes are the result of the adjunction operation on auxiliary trees, α nodes of substitution on initial trees.

coded within the supertags. Usually, a lexical item can have many supertags, depending on the various contexts it appears in. Therefore, the local ambiguity is larger than for the case of POS tags. An LTAG parser for this scenario can be very slow, i.e. its computational complexity is in $O(n^6)$, because of the large number of supertags, i.e. elementary trees, that have to be examined during a parse. In order to speed up the parsing process, we can apply n-gram models on a supertag basis in order to filter out incompatible descriptions and thus improve the performance of the parser. In (Bangalore and Joshi, 1999), a trigram supertagger with smoothing and back-off is reported that achieves an accuracy of 92.2% when trained on one million running words.

There is another aspect to the dependencies coded in the elementary structures. We can use them to actually derive a shallow parse of the sentence in linear time. The procedure is presented in (Bangalore, 2000) and is called *lightweight dependency analysis*. The concept is comparable to *chunking*. The lightweight dependency analyzer (LDA) finds the arguments for the encoded dependency requirements. There exist two types of *slots* that can be filled. On the one hand, nodes marked for substitution (in α-trees) have to be filled by the complements of the lexical anchor. On the other hand, the foot nodes (i.e. nodes marked for adjunction in β-trees) take words that are being modified by the supertag. Figure 1 shows a tree derived by LDA on the sentence *the food was very delicious* from the C-Star'03 corpus (cf. Section 4.1).

The supertagging and LDA tools are available from the XTAG research group website.[2]

As features considered for the reranking experiments we choose:

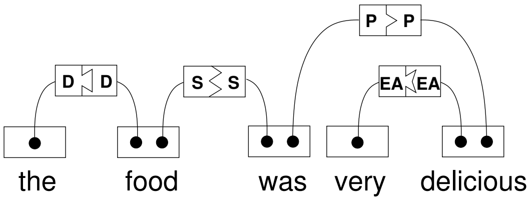

Figure 2: Link grammar: example of a valid linkage satisfying all constraints.

- Supertagger output: directly use the log-likelihoods as feature score. This did not improve performance significantly, so the model was discarded from the final system.
- LDA output:
  - dependency coverage: determine the number of covered elements, i.e. where the dependency slots are filled to the left and right
  - separate features for the number of modifiers and complements determined by the LDA

3.3 Link grammar

Similar to the ideas presented in the previous section, link grammars also explicitly code dependencies between words (Sleator and Temperley, 1993). These dependencies are called *links* which reflect the local requirements of each word. Several constraints have to be satisfied within the link grammar formalism to derive correct linkages, i.e. sets of links, of a sequence of words:

1. Planarity: links are not allowed to cross each other
2. Connectivity: links suffice to connect all words of a sentence
3. Satisfaction: linking requirements of each word are satisfied

An example of a valid linkage is shown in Figure 2. The link grammar parser that we use is freely available from the authors' website.[3] Similar to LTAG, the link grammar formalism is lexicalized which allows for enhancing the methods with probabilistic n-gram models (as is also the case for supertagging). In (Lafferty et al., 1992), the link grammar is used to derive a new class of

[2] `http://www.cis.upenn.edu/~xtag/`

[3] `http://www.link.cs.cmu.edu/link/`

[NP the food] [VP was] [ADJP very delicious]

the/DT food/NN was/VBD very/RB delicious/JJ

Figure 3: Chunking and POS tagging: a tag next to the opening bracket denotes the type of chunk, whereas the corresponding POS tag is given after the word.

language models that, in comparison to traditional n-gram LMs, incorporate capabilities for expressing long-range dependencies between words.

The link grammar dictionary that specifies the words and their corresponding valid links currently holds approximately 60 000 entries and handles a wide variety of phenomena in English. It is derived from newspaper texts.

Within our reranking framework, we use link grammar features that express a possible well-formedness of the translation hypothesis. The simplest feature is a binary one stating whether the link grammar parser could derive a complete linkage or not, which should be a strong indicator of a syntactically correct sentence. Additionally, we added a normalized cost of the matching process which turned out not to be very helpful for rescoring, so it was discarded.

3.4 ME chunking

Like the methods described in the two preceding sections, text chunking consists of dividing a text into syntactically correlated non-overlapping groups of words. Figure 3 shows again our example sentence illustrating this task. Chunks are represented as groups of words between square brackets. We employ the 11 chunk types as defined for the CoNLL-2000 shared task (Tjong Kim Sang and Buchholz, 2000).

For the experiments, we apply a maximum-entropy based tagger which has been successfully evaluated on natural language understanding and named entity recognition (Bender et al., 2003). Within this tool, we directly factorize the posterior probability and determine the corresponding chunk tag for each word of an input sequence. We assume that the decisions depend only on a limited window $e_{i-2}^{i+2} = e_{i-2}...e_{i+2}$ around the current word e_i and on the two predecessor chunk tags c_{i-2}^{i-1}. In addition, part-of-speech (POS) tags g_1^I are assigned and incorporated into the model (cf. Figure 3). Thus, we obtain the following second-order model:

$$
\begin{aligned}
&Pr(c_1^I|e_1^I, g_1^I) = \\
&= \prod_{i=1}^{I} Pr(c_i|c_1^{i-1}, e_1^I, g_1^I) \qquad (4)\\
&= \prod_{i=1}^{I} p(c_i|c_{i-2}^{i-1}, e_{i-2}^{i+2}, g_{i-2}^{i+2}), \qquad (5)
\end{aligned}
$$

where the step from Eq. 4 to 5 reflects our model assumptions.

Furthermore, we have implemented a set of binary valued feature functions for our system, including lexical, word and transition features, prior features, and compound features, cf. (Bender et al., 2003). We run simple count-based feature reduction and train the model parameters using the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff, 1972). In practice, the training procedure tends to result in an overfitted model. To avoid this, a smoothing method is applied where a Gaussian prior on the parameters is assumed (Chen and Rosenfeld, 1999).

Within our reranking framework, we firstly use the ME based tagger to produce the POS and chunk sequences for the different n-best list hypotheses. Given several n-gram models trained on the WSJ corpus for both POS and chunk models, we then rescore the n-best hypotheses and simply use the log-probabilities as additional features. In order to adapt our system to the characteristics of the data used, we build POS and chunk n-gram models on the training corpus part. These domain-specific models are also added to the n-best lists.

The ME chunking approach does not model explicit syntactic linkages of words. Instead, it incorporates a statistical framework to exploit valid and syntactically coherent groups of words by additionally looking at the word classes.

4 Experiments

For the experiments, we use the translation system described in (Zens et al., 2005). Our phrase-based decoder uses several models during search that are interpolated in a log-linear way (as expressed in Eq. 3), such as phrase-based translation models, word-based lexicon models, a language, deletion and simple reordering model and word and phrase penalties. A word graph containing the most likely translation hypotheses is generated during the search process. Out of this compact

		Supplied Data Track			
		Arabic	Chinese	Japanese	English
Train	Sentences	20 000			
	Running Words	180 075	176 199	198 453	189 927
	Vocabulary	15 371	8 687	9 277	6 870
	Singletons	8 319	4 006	4 431	2 888
C-Star’03	Sentences	506			
	Running Words	3 552	3 630	4 130	3 823
	OOVs (Running Words)	133	114	61	65
IWSLT’04	Sentences	500			
	Running Words	3 597	3 681	4 131	3 837
	OOVs (Running Words)	142	83	71	58

Table 1: Corpus statistics after preprocessing.

representation, we extract n-best lists as described in (Zens and Ney, 2005). These n-best lists serve as a starting point for our experiments. The methods presented in Section 3 produce scores that are used as additional features for the n-best lists.

4.1 Corpora

The experiments are carried out on a subset of the *Basic Travel Expression Corpus* (BTEC) (Takezawa et al., 2002), as it is used for the supplied data track condition of the IWSLT evaluation campaign. BTEC is a multilingual speech corpus which contains tourism-related sentences similar to those that are found in phrase books. For the supplied data track, the training corpus contains 20 000 sentences. Two test sets, C-Star’03 and IWSLT’04, are available for the language pairs Arabic-English, Chinese-English and Japanese-English.

The corpus statistics are shown in Table 1. The average source sentence length is between seven and eight words for all languages. So the task is rather limited and very domain-specific. The advantage is that many different reranking experiments with varying feature function settings can be carried out easily and quickly in order to analyze the effects of the different models.

In the following, we use the C-Star’03 set for development and tuning of the system’s parameters. After that, the IWSLT’04 set is used as a blind test set in order to measure the performance of the models.

4.2 Rescoring experiments

The use of n-best lists in machine translation has several advantages. It alleviates the effects of the huge search space which is represented in word graphs by using a compact excerpt of the n best hypotheses generated by the system. Especially for limited domain tasks, the size of the n-best list can be rather small but still yield good oracle error rates. Empirically, n-best lists should have an appropriate size such that the oracle error rate, i.e. the error rate of the best hypothesis with respect to an error measure (such as WER or PER) is approximately half the baseline error rate of the system. N-best lists are suitable for easily applying several rescoring techniques since the hypotheses are already fully generated. In comparison, word graph rescoring techniques need specialized tools which can traverse the graph accordingly. Since a node within a word graph allows for many histories, one can only apply local rescoring techniques, whereas for n-best lists, techniques can be used that consider properties of the whole sentence.

For the Chinese-English and Arabic-English task, we set the n-best list size to $n = 1500$. For Japanese-English, $n = 1000$ produces oracle error rates that are deemed to be sufficiently low, namely 17.7% and 14.8% for WER and PER, respectively. The single-best output for Japanese-English has a word error rate of 33.3% and position-independent word error rate of 25.9%.

For the experiments, we add additional features to the initial models of our decoder that have shown to be particularly useful in the past, such as IBM model 1 score, a clustered language model score and a word penalty that prevents the hypotheses to become too short. A detailed definition of these additional features is given in (Zens et al., 2005). Thus, the baseline we start with is

Chinese → English, C-Star’03	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	8.17	46.2	48.6	41.4
with supertagging/LDA	8.29	46.5	48.4	41.0
with link grammar	8.43	45.6	47.9	41.1
with supertagging/LDA + link grammar	8.22	47.5	47.7	40.8
with ME chunker	8.65	47.3	47.4	40.4
with all models	8.42	47.0	47.4	40.5

Chinese → English, IWSLT’04	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	8.67	45.5	49.1	39.8
with supertagging/LDA	8.68	45.4	49.8	40.3
with link grammar	8.81	45.0	49.0	40.2
with supertagging/LDA+link grammar	8.56	46.0	49.1	40.6
with ME chunker	9.00	44.6	49.3	40.6
with all models	8.89	46.2	48.1	39.6

Table 2: Effect of successively adding syntactic features to the Chinese-English n-best list for C-Star’03 (development set) and IWSLT’04 (test set).

BASE	*Any messages for me?*
RESC	*Do you have any messages for me?*
REFE	*Do you have any messages for me?*
BASE	*She, not yet?*
RESC	*She has not come yet?*
REFE	*Lenny, she has not come in?*
BASE	*How much is it to the?*
RESC	*How much is it to the local call?*
REFE	*How much is it to the city centre?*
BASE	*This blot or.*
RESC	*This is not clean.*
REFE	*This still is not clean.*

Table 3: Translation examples for the Chinese-English test set (IWSLT’04): baseline system (BASE) vs. rescored hypotheses (RESC) and reference translation (REFE).

already a very strong one. The log-linear interpolation weights λ_m from Eq. 3 are directly optimized using the Downhill Simplex algorithm on a linear combination of WER (word error rate), PER (position-independent word error rate), NIST and BLEU score.

In Table 2, we show the effect of adding the presented features successively to the baseline. Separate entries for experiments using supertagging/LDA and link grammars show that a combination of these syntactic approaches always yields some gain in translation quality (regarding BLEU score). The performance of the maximum-entropy based chunking is comparable. A combination of all three models still yields a small improvement.

Table 3 shows some examples for the Chinese-English test set. The rescored translations are syntactically coherent, though semantical correctness cannot be guaranteed. On the test data, we achieve an overall improvement of 0.7%, 0.5% and 0.3% in BLEU score for Chinese-English, Japanese-English and Arabic-English, respectively (cf. Tables 4 and 5).

4.3 Discussion

From the tables, it can be seen that the use of syntactically motivated feature functions within a reranking concept helps to slightly reduce the number of translation errors of the overall translation system. Although the improvement on the IWSLT’04 set is only moderate, the results are nevertheless comparable or better to the ones from (Och et al., 2004), where, starting from IBM model 1 baseline, an additional improvement of only 0.4% BLEU was achieved using more complex methods.

For the maximum-entropy based chunking approach, n-grams with $n = 4$ work best for the chunker that is trained on WSJ data. The domain-specific rescoring model which results from the chunker being trained on the BTEC corpora turns out to prefer higher order n-grams, with $n = 6$ or more. This might be an indicator of the domain-specific rescoring model successfully capturing more local context.

The training of the other models, i.e. supertagging/LDA and link grammar, is also performed on

Japanese → English, C-Star'03	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	9.09	57.8	31.3	25.0
with supertagging/LDA	9.13	57.8	31.3	24.8
with link grammar	9.46	57.6	31.9	25.3
with supertagging/LDA + link grammar	9.24	58.2	31.0	24.8
with ME chunker	9.31	58.7	30.9	24.4
with all models	9.21	58.9	30.5	24.3

Japanese → English, IWSLT'04	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	9.22	54.7	34.1	25.5
with supertagging/LDA	9.27	54.8	34.2	25.6
with link grammar	9.37	54.9	34.3	25.9
with supertagging/LDA + link grammar	9.30	55.0	34.0	25.6
with ME chunker	9.27	55.0	34.2	25.5
with all models	9.27	55.2	33.9	25.5

Table 4: Effect of successively adding syntactic features to the Japanese-English n-best list for C-Star'03 (development set) and IWSLT'04 (test set).

Arabic → English, C-Star'03	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	10.18	64.3	23.9	20.6
with supertagging/LDA	10.13	64.6	23.4	20.1
with link grammar	10.06	64.7	23.4	20.3
with supertagging/LDA + link grammar	10.20	65.0	23.2	20.2
with ME chunker	10.11	65.1	23.0	19.9
with all models	10.23	65.2	23.0	19.9

Arabic → English, IWSLT'04	NIST	BLEU[%]	mWER[%]	mPER[%]
Baseline	9.75	59.8	26.1	21.9
with supertagging/LDA	9.77	60.5	25.6	21.5
with link grammar	9.74	60.5	25.9	21.7
with supertagging/LDA + link grammar	9.86	60.8	26.0	21.6
with ME chunker	9.71	59.9	25.9	21.8
with all models	9.84	60.1	26.4	21.9

Table 5: Effect of successively adding syntactic features to the Arabic-English n-best list for C-Star'03 (development set) and IWSLT'04 (test set).

out-of-domain data. Thus, further improvements should be possible if the models were adapted to the BTEC domain. This would require the preparation of an annotated corpus for the supertagger and a specialized link grammar, which are both time-consuming tasks.

The syntactically motivated methods (supertagging/LDA and link grammars) perform similarly to the maximum-entropy based chunker. It seems that both approaches successfully exploit structural properties of language. However, one outlier is ME chunking on the Chinese-English test data, where we observe a lower BLEU but a larger NIST score. For Arabic-English, the combination of all methods does not seem to generalize well on the test set. In that case, supertagging/LDA and link grammar outperforms the ME chunker: the overall improvement is 1% absolute in terms of BLEU score.

5 Conclusion

We added syntactically motivated features to a statistical machine translation system in a reranking framework. The goal was to analyze whether shallow parsing techniques help in identifying ungrammatical hypotheses. We showed that some improvements are possible by utilizing supertagging, lightweight dependency analysis, a link grammar parser and a maximum-entropy based chunk parser. Adding features to n-best lists and discriminatively training the system on a development set helped to gain up to 0.7% in BLEU score on the test set.

Future work could include developing an adapted LTAG for the BTEC domain or incorporating n-gram models into the link grammar concept in order to derive a long-range language model (Lafferty et al., 1992). However, we feel that the current improvements are not significant enough to justify these efforts. Additionally, we will apply these reranking methods to larger corpora in order to study the effects on longer sentences from more complex domains.

Acknowledgments

This work has been partly funded by the European Union under the integrated project TC-Star (Technology and Corpora for Speech to Speech Translation, IST-2002-FP6-506738, http://www.tc-star.org), and by the R&D project TRAMES managed by Bertin Technologies as prime contractor and operated by the french DGA (Délégation Générale pour l'Armement).

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Srinivas Bangalore. 2000. A lightweight dependency analyzer for partial parsing. *Computational Linguistics*, 6(2):113–138.

Oliver Bender, Klaus Macherey, Franz Josef Och, and Hermann Ney. 2003. Comparison of alignment templates and maximum entropy models for natural language understanding. In *EACL03: 10th Conf. of the Europ. Chapter of the Association for Computational Linguistics*, pages 11–18, Budapest, Hungary, April.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon University, Pittsburgh, PA.

J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480.

John Lafferty, Daniel Sleator, and Davy Temperley. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Proc. of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 89–97, Cambridge, MA.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proc. 2004 Meeting of the North American chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 161–168, Boston, MA.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Daniel Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*, Tilburg/Durbuy, The Netherlands/Belgium, August.

Toshiyuki Takezawa, Eiichiro Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of the Third Int. Conf. on Language Resources and Evaluation (LREC)*, pages 147–152, Las Palmas, Spain, May.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, Lisbon, Portugal, September.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania, Philadelphia, PA, USA.

Richard Zens and Hermann Ney. 2005. Word graphs for statistical machine translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 191–198, Ann Arbor, MI, June.

Richard Zens, Oliver Bender, Saša Hasan, Shahram Khadivi, Evgeny Matusov, Jia Xu, Yuqi Zhang, and Hermann Ney. 2005. The RWTH phrase-based statistical machine translation system. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 155–162, Pittsburgh, PA, October.