CCG Supertags in Factored Statistical Machine Translation

Alexandra Birch a.c.birch-mayne@sms.ed.ac.uk
Miles Osborne miles@inf.ed.ac.uk
Philipp Koehn pkoehn@inf.ed.ac.uk
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh, EH8 9LW, UK

Abstract

Combinatorial Categorial Grammar (CCG) supertags present phrase-based machine translation with an opportunity to access rich syntactic information at a word level. The challenge is incorporating this information into the translation process. Factored translation models allow the inclusion of supertags as a factor in the source or target language. We show that this results in an improvement in the quality of translation and that the value of syntactic supertags in flat structured phrase-based models is largely due to better local reorderings.

1 Introduction

In large-scale machine translation evaluations, phrase-based models generally outperform syntax-based models[1]. Phrase-based models are effective because they capture the lexical dependencies between languages. However, these models, which are equivalent to finite-state machines (Kumar and Byrne, 2003), are unable to model long range word order differences. Phrase-based models also lack the ability to incorporate the generalisations implicit in syntactic knowledge and they do not respect linguistic phrase boundaries. This makes it difficult to improve reordering in phrase-based models.

Syntax-based models can overcome some of the problems associated with phrase-based models because they are able to capture the long range structural mappings that occur in translation. Recently there have been a few syntax-based models that show performance comparable to the phrase-based models (Chiang, 2005; Marcu et al., 2006). However, reliably learning powerful rules from parallel data is very difficult and prone to problems with sparsity and noise in the data. These models also suffer from a large search space when decoding with an integrated language model, which can lead to search errors (Chiang, 2005).

In this paper we investigate the idea of incorporating syntax into phrase-based models, thereby leveraging the strengths of both the phrase-based models and syntactic structures. This is done using CCG supertags, which provide a rich source of syntactic information. CCG contains most of the structure of the grammar in the lexicon, which makes it possible to introduce CCG supertags as a factor in a factored translation model (Koehn et al., 2006). Factored models allow words to be vectors of features: one factor could be the surface form and other factors could contain linguistic information.

Factored models allow for the easy inclusion of supertags in different ways. The first approach is to generate CCG supertags as a factor in the target and then apply an n-gram model over them, increasing the probability of more frequently seen sequences of supertags. This is a simple way of including syntactic information in a phrase-based model, and has also been suggested by Hassan et al. (2007). For both Arabic-English (Hassan et al., 2007) and our experiments in Dutch-English, n-gram models over CCG supertags improve the quality of translation. By preferring more likely sequences of supertags, it is conceivable that the output of the decoder is

[1] www.nist.gov/speech/tests/mt/mt06eval_official_results.html

Proceedings of the Second Workshop on Statistical Machine Translation, pages 9–16,
Prague, June 2007.

more grammatical. However, its not clear exactly how syntactic information can benefit a flat structured model: the constraints contained within supertags are not enforced and relationships between supertags are not linear. We perform experiments to explore the nature and limits of the contribution of supertags, using different orders of n-gram models, reordering models and focussed manual evaluation. It seems that the benefit of using n-gram supertag sequence models is largely from improving reordering, as much of the gain is eroded by using a lexicalised reordering model. This is supported by the manual evaluation which shows a 44% improvement in reordering Dutch-English verb final sentences.

The second and novel way we use supertags is to direct the translation process. Supertags on the source sentence allows the decoder to make decisions based on the structure of the input. The subcategorisation of a verb, for instance, might help select the correct translation. Using multiple dependencies on factors in the source, we need a strategy for dealing with sparse data. We propose using a logarithmic opinion pool (Smith et al., 2005) to combine the more specific models (which depend on both words and supertags) with more general models (which only depends on words). This paper is the first to suggest this approach for combining multiple information sources in machine translation.

Although the addition of supertags to phrase-based translation does show some improvement, their overall impact is limited. Sequence models over supertags clearly result in some improvements in local reordering but syntactic information contains long distance dependencies which are simply not utilised in phrase-based models.

2 Factored Models

Inspired by work on factored language models, Koehn et al. (2006) extend phrase-based models to incorporate multiple levels of linguistic knowledge as factors. Phrase-based models are limited to sequences of words as their units with no access to additional linguistic knowledge. Factors allow for richer translation models, for example, the gender or tense of a word can be expressed. Factors also allow the model to generalise, for example, the lemma of a word could be used to generalise to unseen inflected forms.

The factored translation model combines features in a log-linear fashion (Och, 2003). The most likely target sentence $\hat{t}$ is calculated using the decision rule in Equation 1:

$$\hat{t} = \arg\max_{t} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^{F_s}, t_1^{F_t}) \right\} \quad (1)$$

$$\hat{t} \propto \sum_{m=1}^{M} \lambda_m h_m(s_1^{F_s}, t_1^{F_t}) \quad (2)$$

where M is the number of features, $h_m(s_1^{F_s}, t_1^{F_t})$ are the feature functions over the factors, and λ are the weights which combine the features which are optimised using minimum error rate training (Venugopal and Vogel, 2005). Each function depends on a vector $s_1^{F_s}$ of source factors and a vector $t_1^{F_t}$ of target factors. An example of a factored model used in upcoming experiments is:

$$\hat{t} \propto \sum_{m=1}^{M} \lambda_m h_m(s_w, t_{wc}) \quad (3)$$

where s_w means the model depends on (s)ource (w)ords, and t_{wc} means the model generates (t)arget (w)ords and (c)cg supertags. The model is shown graphically in Figure 1.

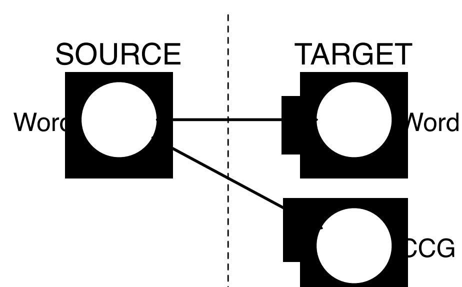

Figure 1. Factored translation with source words determining target words and CCG supertags

For our experiments we used the following features: the translation probabilities $Pr(s_1^{F_s}|t_1^{F_t})$ and $Pr(t_1^{F_t}|s_1^{F_s})$, the lexical weights (Koehn et al., 2003) $lex(s_1^{F_s}|t_1^{F_t})$ and $lex(t_1^{F_t}|s_1^{F_s})$, and a phrase penalty e, which allows the model to learn a preference for longer or shorter phrases. Added to these features is the word penalty e^{-1} which allows the model to learn a preference for longer or shorter sentences, the distortion model d that prefers monotone word order, and the language model probability $Pr(t)$. All these features are logged when combined in the log-linear model in order to retain the impact of very unlikely translations or sequences.

One of the strengths of the factored model is it allows for n-gram distributions over factors on the target. We call these distributions *sequence models*. By analogy with language models, for example, we can construct a bigram sequence model as follows:

$$p(f_1, f_2, \ldots f_n) = p(f_1) \prod_{i=2}^{n} p(f_i | f_{(i-1)})$$

where f is a factor (eg. CCG supertags) and n is the length of the string. Sequence models over POS tags or supertags are smaller than language models because they have restricted lexicons. Higher order, more powerful sequence models can therefore be used.

Applying multiple factors in the source can lead to sparse data problems. One solution is to break down the translation into smaller steps and translate each factor separately like in the following model where source words are translated separately to the source supertags:

$$\hat{t} \propto \sum_{m=1}^{M} \lambda_m h_m(s_w, t_w) + \sum_{n=1}^{N} \lambda_n h_n(s_c, t_w)$$

However, in many cases multiple dependencies are desirable. For instance translating CCG supertags independently of words could introduce errors. Multiple dependencies require some form of backing off to simpler models in order to cover the cases where, for instance, the word has been seen in training, but not with that particular supertag. Different backoff paths are possible, and it would be interesting but prohibitively slow to apply a strategy similar to generalised parallel backoff (Bilmes and Kirchhoff, 2003) which is used in factored language models. Backoff in factored language models is made more difficult because there is no obvious backoff path. This is compounded for factored phrase-based translation models where one has to consider backoff in terms of factors and n-gram lengths in both source and target languages. Furthermore, the surface form of a word is probably the most valuable factor and so its contribution must always be taken into account. We therefore did not use backoff and chose to use a log-linear combination of features and models instead.

Our solution is to extract two translation models:

$$\hat{t} \propto \sum_{m=1}^{M} \lambda_m h_m(s_{wc}, t_w) + \sum_{n=1}^{N} \lambda_n h_n(s_w, t_w) \quad (4)$$

One model consists of more specific features m and would return log probabilities, for example $log_2 Pr(t_w | s_{wc})$, if the particular word and supertag had been seen before in training. Otherwise it returns $-C$, a negative constant emulating $log_2(0)$. The other model consist of more general features n and always returns log probabilities, for example $log_2 Pr(t_w | s_w)$.

3 CCG and Supertags

CCGs have syntactically rich lexicons and a small set of combinatory operators which assemble the parse-trees. Each word in the sentence is assigned a category from the lexicon. A category may either be atomic (**S**, **NP** etc.) or complex (**S\S**, **(S\NP)/NP** etc.). Complex categories have the general form α/β or $\alpha\backslash\beta$ where α and β are themselves categories. An example of a CCG parse is given:

$$\begin{array}{ccc}
\text{Peter} & \text{eats} & \text{apples} \\
\overline{\mathbf{NP}} & \overline{\mathbf{(S\backslash NP)/NP}} & \overline{\mathbf{NP}} \\
 & \multicolumn{2}{c}{\overline{\qquad\qquad\qquad}^{>}} \\
 & \multicolumn{2}{c}{\mathbf{S\backslash NP}} \\
\multicolumn{3}{c}{\overline{\qquad\qquad\qquad\qquad\qquad}^{<}} \\
\multicolumn{3}{c}{\mathbf{S}}
\end{array}$$

where the derivation proceeds as follows: “eats” is combined with “apples” under the operation of forward application. “eats” can be thought of as a function that takes a **NP** to the right and returns a **S\NP**. Similarly the phrase “eats apples” can be thought of as a function which takes a noun phrase **NP** to the left and returns a sentence **S**. This operation is called backward application.

A sentence together with its CCG categories already contains most of the information present in a full parse. Because these categories are lexicalised, they can easily be included into factored phrase-based translation. CCG supertags are categories that have been provided by a supertagger. Supertags were introduced by Bangalore (1999) as a way of increasing parsing efficiency by reducing the number of structures assigned to each word. Clark (2002) developed a suppertagger for CCG which uses a conditional maximum entropy model to estimate the probability of words being assigned particular categories. Here is an example of a sentence that has been supertagged in the training corpus:

We	all	agree	on	that	.
NP	**NP\NP**	**(S[dcl]\NP)/PP**	**PP/NP**	**NP**	.

The verb "agree" has been assigned a complex supertag **(S[dcl]\NP)/PP** which determines the type and direction of its arguments. This information can be used to improve the quality of translation.

4 Experiments

The first set of experiments explores the effect of CCG supertags on the target, translating from Dutch into English. The last experiment shows the effect of CCG supertags on the source, translating from German into English. These language pairs present a considerable reordering challenge. For example, Dutch and German have SOV word order in subordinate clauses. This means that the verb often appears at the end of the clause, far from the position of the English verb.

4.1 Experimental Setup

The experiments were run using Moses[2], an open source factored statistical machine translation system. The SRILM language modelling toolkit (Stolcke, 2002) was used with modified Kneser-Ney discounting and interpolation. The CCG supertagger (Clark, 2002; Clark and Curran, 2004) was provided with the C&C Language Processing Tools[3]. The supertagger was trained on the CCGBank in English (Hockenmaier and Steedman, 2005) and in German (Hockenmaier, 2006).

The Dutch-English parallel training data comes from the Europarl corpus (Koehn, 2005) and excludes the proceedings from the last quarter of 2000. This consists of 855,677 sentences with a maximum of 50 words per sentence. 500 sentences of tuning data and the 2000 sentences of test data are taken from the ACL Workshop on Building and Using Parallel Texts[4].

The German-English experiments use data from the NAACL 2006 Workshop on Statistical Machine Translation[5]. The data consists of 751,088 sentences of training data, 500 sentences of tuning data and 3064 sentences of test data. The English and German training sets were POS tagged and supertagged before lowercasing. The language models and the sequence models were trained on the Europarl training data. Where not otherwise specified, the POS tag and supertag sequence models are 5-gram models and the language model is a 3-gram model.

4.2 Sequence Models Over Supertags

Our first Dutch-English experiment seeks to establish what effect sequence models have on machine translation. We show that supertags improve translation quality. Together with Shen et al. (2006) it is one of the first results to confirm the potential of the factored model.

Model	BLEU
s_w, t_w	23.97
s_w, t_{wp}	24.11
s_w, t_{wc}	24.42
s_w, t_{wpc}	**24.43**

Table 1. The effect of sequence models on Dutch-English BLEU score. Factors are (w)ords, (p)os tags, (c)cg supertags on the source s or the target t

Table 1 shows that sequence models over CCG supertags in the target (model s_w, t_{wc}) improves over the baseline (model s_w, t_w) which has no supertags. Supertag sequence models also outperform models which apply POS tag sequence models (s_w, t_{wp}) and, interestingly do just as well as models which apply both POS tag and supertag sequence models (s_w, t_{wps}). Supertags are more informative than POS tags as they contain the syntactic context of a word.

These experiments were run with the distortion limit set to 6. This means that at most 6 words in

[2]see http://www.statmt.org/moses/

[3]see http://svn.ask.it.usyd.edu.au/trac/candc/wiki

[4]see http://www.statmt.org/wpt05/

[5]see http://www.statmt.org/wpt06/

the source sentence can be skipped. We tried setting the distortion limit to 15 to see if allowing longer distance reorderings with CCG supertag sequence models could further improve performance, however it resulted in a decrease in performance to a BLEU score of 23.84.

4.3 Manual Analysis

The BLEU score improvement in Table 1 does not explain how the supertag sequence models affect the translation process. As suggested by Callison-Burch et al.(2006) we perform a focussed manual analysis of the output to see what changes have occurred.

From the test set, we randomly selected 100 sentences which required reordering of verbs: the Dutch sentences ended with a verb which had to be moved forward in the English translation. We record whether or not the verb was correctly translated and whether it was reordered to the correct position in the target sentence.

Model	Translated	Reordered
s_w, t_w	81	36
s_w, t_{wc}	87	52

Table 2. Analysis of % correct translation and reordering of verbs for Dutch-English translation

In Table 2 we can see that the addition of the CCG supertag sequence model improved both the translation of the verbs and their reordering. However, the improvement is much more pronounced for reordering. The difference in the reordering results is significant at $p < 0.05$ using the χ^2 significance test. This shows that the syntactic information in the CCG supertags is used by the model to prefer better word order for the target sentence.

In Figure 2 we can see two examples of Dutch-English translations that have improved with the application of CCG supertag sequence models. In the first example the verb "heeft" occurs at the end of the source sentence. The baseline model (s_w, t_w) does not manage to translate "heeft". The model with the CCG supertag sequence model (s_w, t_{wc}) translates it correctly as "has" and reorders it correctly 4 places to the left. The second example also shows the sequence model correctly translating the Dutch verb at the end of the sentence "nodig". One can see that it is still not entirely grammatical.

The improvements in reordering shown here are reorderings over a relatively short distance, two or three positions. This is well within the 5-gram order of the CCG supertag sequence model and we therefore consider this to be local reordering.

4.4 Order of the Sequence Model

The CCG supertags describe the syntactic context of the word they are attached to. Therefore they have an influence that is greater in scope than surface words or POS tags. Increasing the order of the CCG supertag sequence model should also increase the ability to perform longer distance reordering. However, at some point the reliability of the predictions of the sequence models is impaired due to sparse counts.

Model	None	1gram	3gram	5gram	7gram
s_w, t_{wc}	24.18	23.96	24.19	**24.42**	24.32
s_w, t_{wpc}	24.34	23.86	24.09	**24.43**	24.14

Table 3. BLUE scores for Dutch-English models which apply CCG supertag sequence models of varying orders

In Table 3 we can see that the optimal order for the CCG supertag sequence models is 5.

4.5 Language Model vs. Supertags

The language model makes a great contribution to the correct order of the words in the target sentence. In this experiment we investigate whether by using a stronger language model the contribution of the sequence model will no longer be relevant. The relative contribution of the language mode and different sequence models is investigated for different language model n-gram lengths.

Model	None	1gram	3gram	5gram	7gram
s_w, t_w	-	21.22	23.97	24.05	24.13
s_w, t_{wp}	21.87	21.83	24.11	24.25	24.06
s_w, t_{wc}	21.75	21.70	24.42	**24.67**	**24.60**
s_w, t_{wpc}	**21.99**	**22.07**	**24.43**	24.48	24.42

Table 4. BLEU scores for Dutch-English models which use language models of increasing n-gram length. Column None does not apply any language model. Model s_w, t_w does not apply any sequence models, and model s_w, t_{wpc} applies both POS tag and supertag sequence models.

In Table 4 we can see that if no language model is present(None), the system benefits slightly from

source:hij kan toch niet beweren dat hij daar geen exacte informatie over **heeft** !
reference: how can he say he does not **have** any precise information ?
s_w, t_w:he cannot say that he is not an exact information about .
s_w, t_{wc}: he cannot say that he **has** no precise information on this !
source: wij moeten hun verwachtingen niet beschamen . meer dan ooit hebben al die landen thans onze bijstand **nodig**
reference: we must not disappoint them in their expectations , and now more than ever these countries **need** our help
s_w, t_w:we must not fail to their expectations , more than ever to have all these countries now our assistance **necessary**
s_w, t_{wc}: we must not fail to their expectations , more than ever , those countries now **need** our assistance

Figure 2. Examples where the CCG supertag sequence model improves Dutch-English translation

having access to all the other sequence models. However, the language model contribution is very strong and in isolation contributes more to translation performance than any other sequence model. Even with a high order language model, applying the CCG supertag sequence model still seems to improve performance. This means that even if we use a more powerful language model, the structural information contained in the supertags continues to be beneficial.

4.6 Lexicalised Reordering vs. Supertags

In this experiment we investigate using a stronger reordering model to see how it compares to the contribution that CCG supertag sequence models make. Moses implements the lexicalised reordering model described by Tillman (2004), which learns whether phrases prefer monotone, inverse or disjoint orientations with regard to adjacent phrases. We apply this reordering models to the following experiments.

Model	None	Lex. Reord.
s_w, t_w	23.97	24.72
s_w, t_{wc}	24.42	24.78

Table 5. Dutch-English models with and without a lexicalised reordering model.

In Table 5 we can see that lexicalised reordering improves translation performance for both models. However, the improvement that was seen using CCG supertags without lexicalised reordering, almost disappears when using a stronger reordering model. This suggests that CCG supertags' contribution is similar to that of a reordering model. The lexicalised reordering model only learns the orientation of a phrase with relation to its adjacent phrase, so its influence is very limited in range. If it can replace CCG supertags, it suggests that supertags' influence is also within a local range.

4.7 CCG Supertags on Source

Sequence models over supertags improve the performance of phrase-based machine translation. However, this is a limited way of leveraging the rich syntactic information available in the CCG categories. We explore the potential of letting supertags direct translation by including them as a factor on the source. This is similar to syntax-directed translation originally proposed for compiling (Aho and Ullman, 1969), and also used in machine translation (Quirk et al., 2005; Huang et al., 2006). Information about the source words' syntactic function and subcategorisation can directly influence the hypotheses being searched in decoding. These experiments were performed on the German to English translation task, in contrast to the Dutch to English results given in previous experiments.

We use a model which combines more specific dependencies on source words and source CCG supertags, with a more general model which only has dependancies on the source word, see Equation 4. We explore two different ways of balancing the statistical evidence from these multiple sources. The first way to combine the general and specific sources of information is by considering features from both models as part of one large log-linear model. However, by including more and less informative features in one model, we may transfer too much explanatory power to the more specific features. To overcome this problem, Smith et al. (2006) demonstrated that using ensembles of separately trained models and combining them in a logarithmic opinion pool (LOP) leads to better parameter values. This approach was used as the second way in which we combined our models. An ensemble of log-linear models was combined using a multiplicative constant γ which we train manually using held out data.

$$\hat{t} \propto \sum_{m=1}^{M} \lambda_m h_m(s_{wc}, t_w) + \gamma \left(\sum_{n=1}^{N} \lambda_n h_n(s_w, t_w) \right)$$

Typically, the two models would need to be normalised before being combined, but here the multiplicative constant fulfils this rôle by balancing their separate contributions. This is the first work suggesting the application of LOPs to decoding in machine translation. In the future more sophisticated translation models and ensembles of models will need methods such as LOPs in order to balance statistical evidence from multiple sources.

Model	BLEU
s_w, t_w	23.30
s_{wc}, t_w	19.73
single	23.29
LOP	23.46

Table 6. German-English: CCG supertags are used as a factor on the source. The simple models are combined in two ways: either as a single log-linear model or as a LOP of log-linear models

Table 6 shows that the simple, general model (model s_w, t_w) performs considerably better than the simple specific model, where there are multiple dependencies on both words and CCG supertags (model s_{wc}, t_w). This is because there are words in the test sentence that have been seen before but not with the CCG supertag. Statistical evidence from multiple sources must be combined. The first way to combine them is to join them in one single log-linear model, which is trained over many features. This makes finding good weights difficult as the influence of the general model is greater, and its difficult for the more specific model to discover good weights. The second method for combining the information is to use the weights from the separately trained simple models and then combine them in a LOP. Held out data is used to set the multiplicative constant needed to balance the contribution of the two models. We can see that this second approach is more successful and this suggests that it is important to carefully consider the best ways of combining different sources of information when using ensembles of models. However, the results of this experiment are not very conclusive. There is no uncertainty in the source sentence and the value of modelling it using CCG supertags is still to be demonstrated.

5 Conclusion

The factored translation model allows for the inclusion of valuable sources of information in many different ways. We have shown that the syntactically rich CCG supertags do improve the translation process and we investigate the best way of including them in the factored model. Using CCG supertags over the target shows the most improvement, especially when using targeted manual evaluation. However, this effect seems to be largely due to improved local reordering. Reordering improvements can perhaps be more reliably made using better reordering models or larger, more powerful language models. A further consideration is that supertags will always be limited to the few languages for which there are treebanks.

Syntactic information represents embedded structures which are naturally incorporated into grammar-based models. The ability of a flat structured model to leverage this information seems to be limited. CCG supertags' ability to guide translation would be enhanced if the constraints encoded in the tags were to be enforced using combinatory operators.

6 Acknowledgements

We thank Hieu Hoang for assistance with Moses, Julia Hockenmaier for access to CCGbank lexicons in German and English, and Stephen Clark and James Curran for providing the supertagger. This work was supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022 and in part under the EuroMatrix project funded by the European Commission (6th Framework Programme).

References

Alfred V. Aho and Jeffrey D. Ullman. 1969. Properties of syntax directed translations. *Journal of Computer and System Sciences*, 3(3):319–334.

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

Jeff Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the North American Association for Computational Linguistics Conference*, Edmonton, Canada.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan.

Stephen Clark and James R. Curran. 2004. Parsing the wsj using ccg and log-linear models. In *Proceedings of the Association for Computational Linguistics*, pages 103–110, Barcelona, Spain.

Stephen Clark. 2002. Supertagging for combinatory categorial grammar. In *Proceedings of the International Workshop on Tree Adjoining Grammars*, pages 19–24, Venice, Italy.

Hany Hassan, Khalil Sima'an, and Andy Way. 2007. Supertagged phrase-based statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, Prague, Czech Republic. (to appear).

Julia Hockenmaier and Mark Steedman. 2005. Ccgbank manual. Technical Report MS-CIS-05-09, Department of Computer and Information Science, University of Pennsylvania.

Julia Hockenmaier. 2006. Creating a ccgbank and a wide-coverage ccg lexicon for german. In *Proceedings of the International Conference on Computational Linguistics and of the Association for Computational Linguistics*, Sydney, Australia.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A syntax-directed translator with extended domain of locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8, New York City, New York. Association for Computational Linguistics.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pages 127–133, Edmonton, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Richard Zens, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2006. Open source toolkit for statistical machine translation. In *Summer Workshop on Language Engineering, John Hopkins University Center for Language and Speech Processing*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Shankar Kumar and William Byrne. 2003. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pages 63–70, Edmonton, Canada.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the Association for Computational Linguistics*, pages 271–279, Ann Arbor, Michigan.

Wade Shen, Richard Zens, Nicola Bertoldi, and Marcello Federico. 2006. The JHU workshop 2006 IWSLT system. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 59–63, Kyoto, Japan.

Andrew Smith and Miles Osborne. 2006. Using gazetteers in discriminative information extraction. In *The Conference on Natural Language Learning*, New York City, USA.

Andrew Smith, Trevor Cohn, and Miles Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *Proceedings of the Association for Computational Linguistics*, pages 18–25, Ann Arbor, Michigan.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of Spoken Language Processing*, pages 901–904.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pages 101–104, Boston, USA. Association for Computational Linguistics.

Ashish Venugopal and Stephan Vogel. 2005. Considerations in MCE and MMI training for statistical machine translation. In *Proceedings of the European Association for Machine Translation*, Budapest, Hungary.