EXPLORING THE PARAMETER SPACE IN STATISTICAL MACHINE TRANSLATION VIA F-STRUCTURE TRANSFER

Yvette Graham and Josef van Genabith
Centre for Next Generation Localisation

Proceedings of the LFG12 Conference

Miriam Butt and Tracy Holloway King (Editors)

2012

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

Machine translation can be carried out via transfer between source and target language deep syntactic structures. In this paper, we examine core parameters of such a system in the context of a statistical approach where the translation model, based on deep syntax, is automatically learned from parsed bilingual corpora. We provide a detailed empirical investigation into the effects of core parameters on translation quality for the German-English translation pair, such as methods of word alignment, limits on the size of transfer rules, transfer decoder beam size, n-best target input representations for generation, as well as deterministic versus non-deterministic generation. Results highlight just how vital employing a suitable method of word alignment is for this approach as well as the significant trade-off between gains in Bleu score and increase in overall translation time that exists when n-best structures are generated.

1 Introduction

Statistical Machine Translation via deep syntactic transfer is carried out in three steps: (i) parsing the source language (SL) input to SL deep syntactic representation, (ii) transfer from SL deep syntactic representation to target language (TL) deep syntactic representation, (iii) generation of TL string. Figure 1 shows how an example German sentence is translated into English. Bojar and Hajič (2008) present an English to Czech SMT system that uses the Functional Generative Description (FGD) (Sgall et al., 1986) Tectogrammatical Layer (T-layer), i.e. labeled ordered dependency trees, as intermediate representation for transfer, and integrate a bigram dependency-based language model into decoding. Riezler and Maxwell (Riezler and Maxwell, 2006) use the Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001) functional structure (f-structure) for transfer, an attribute-value structure encoding of bilexical labeled dependencies and atomic value features, and extract transfer rules semi-automatically from the training data, by automatically word aligning surface-form sentences using Giza++ (Och et al., 1999) before manually detecting and automatically correcting systematic errors. Most of the transfer rules are automatically extracted from the parsed training data with some transfer rules manually written and deep syntax language modeling is carried out after decoding, on the n-best output structures.[1]

Like Riezler and Maxwell (2006), we use the LFG f-structure as the intermediate representation for transfer, but in contrast we investigate the feasibility of deep syntactic transfer when translation models are learned fully automatically. In addition, we integrate a deep syntax language model to decoder search, similar to Bojar and Hajič (2008) but increase to a tri-gram model. Again in contrast to Riezler and Maxwell (2006) where language modeling is applied to the n-best structures output

[†]This work was partly funded by a Science Foundation Ireland PhD studentship P07077-60101.

[1]Personal communication with authors.

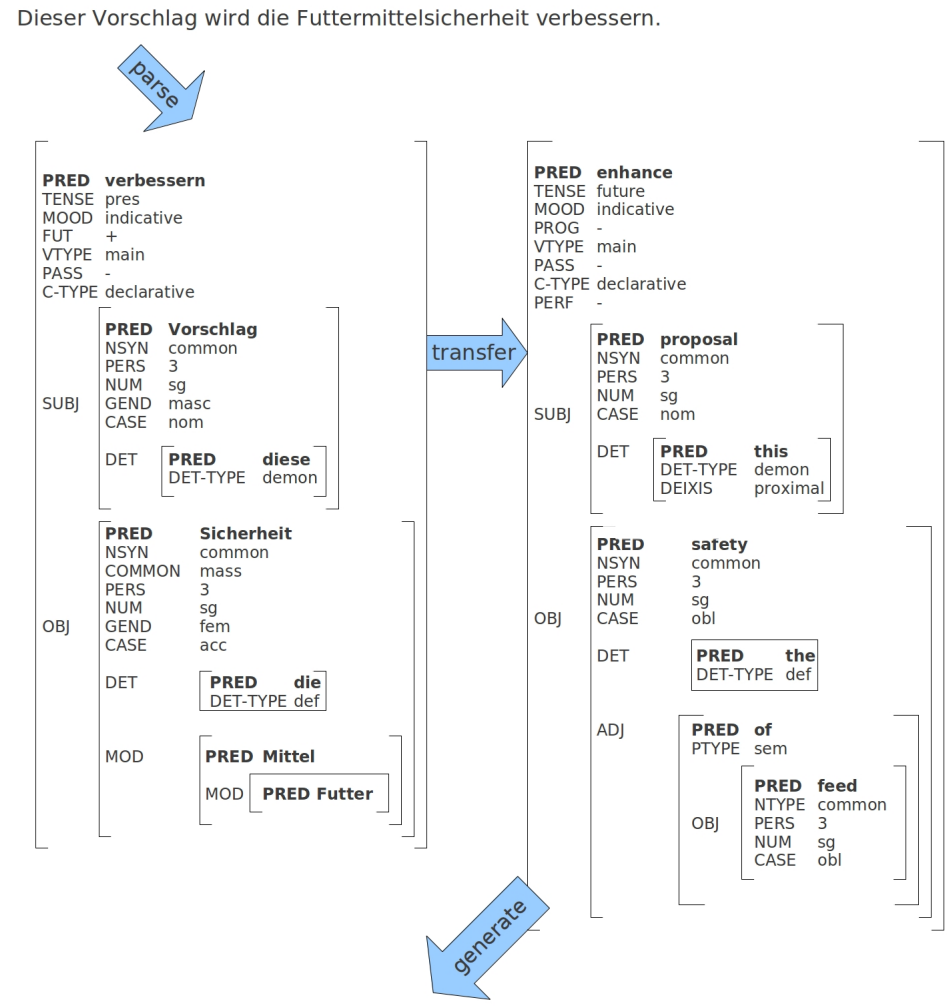


Figure 1: Deep syntactic transfer example via LFG f-structures

after decoding, we integrate language modeling to decoder search. Our empirical evaluation highlights the importance of selecting methods of word alignment most suitable for deep syntax, as well as notable trade-offs that exist between currently achievable translation speed and the quality of translations produced.

Ding and Palmer (2006) use dependency structures for translation, but the approach they take is not strictly deep syntactic transfer, as they use dependency relations between surface form words as opposed to lemmas and morpho-syntactic information, and additionally they use information about source language word order during translation, arguably losing the high level of language pair independence afforded by fully deep syntactic transfer.

2 Translation Model

Similar to PB-SMT (Koehn et al., 2003), our translation model is a log-linear combination of several feature functions:

$$p(e|f) = exp \sum_{i=1}^{n} \lambda_i h_i(e, f) \quad (1)$$

2.1 Word Alignment

An alignment between the nodes of the SL and TL deep syntactic training structures is required in order to automatically extract transfer rules. In our evaluation, we investigate the following three methods of word (or node) alignment, all using Giza++ (Och et al., 1999) for alignment and Moses (Koehn et al., 2007) for symmetrization:

- SF-GDF: input the surface-form bitext corpus to Giza++ and symmetrize with grow-diag-final algorithm.[2] Map many-to-many word alignment from each surface-form word to its corresponding local f-structure. This yields a many-to-many alignment between local f-structures and was used in Riezler and Maxwell (2006).[3]

- DS-INT: reconstruct a bitext corpus by extracting predicates from each local f-structure, input the reconstructed bitext to Giza++, then use the intersection of the bidirectional word alignment for symmetrization. This yields a one-to-one alignment between local f-structures. This method takes advantage of the predicate values of f-structures being in the more general lemma form, and should suffer less from data sparseness problems.

[2]Grow-diag-final works as follows: Word alignment is run in both language directions, for example, German-to-English (f2e) and English-to-German (e2f). For any given training sentence pair, each run (e2f and f2e) can yield a different set of alignment points between the words of the training sentence pair. There are many ways to combine these two sets, grow-diag-final begins with the intersection, then adds unaligned words.

[3]It should be noted that we use a different method of transfer rule extraction, we do not correct word alignment and do not include hand-crafted transfer rules.

- DS-GDF: reconstruct a bitext corpus by extracting predicates from each local f-structure, input the reconstructed bitext to Giza++ (as in DS-INT), but use grow-diag-final for symmetrization yielding up to many-to-many alignments between local f-structures.

2.2 Transfer Rule Extraction

Similar to PB-SMT, the transfer of a SL deep syntactic structure **f** into a TL deep syntactic structure **e** can be broken down into the transfer of a set of rules $\{\bar{f}, \bar{e}\}$:

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) \tag{2}$$

In PB-SMT, all phrases consistent with the word alignment are extracted, with shorter phrases needed for high coverage of unseen data and larger phrases improving TL fluency (Koehn et al., 2003). With the same motivation, we extract all transfer rules consistent with the node alignment. Figure 2 shows a subset of the transfer rules extracted from the f-structure pair in Figure 1.[4] We estimate the translation probability distribution using relative frequencies of transfer rules:

$$\phi(\bar{f}, \bar{e}) = \frac{count(\bar{e}, \bar{f})}{\sum_{\bar{f}_i} count(\bar{e}, \bar{f}_i)} \tag{3}$$

This is carried out in both the source-to-target and target-to-source directions.[5]

3 Deep Syntax Language Model

In deep syntactic transfer, the output of the decoder is a TL deep syntactic structure with words organized in the form of a graph (as opposed to a linear sequence of words in PB-SMT). A standard surface-form language model cannot be used during transfer decoding because no surface-form representation of the TL deep syntactic structure is available. It is still important for the model to take TL fluency into account so that the structures it outputs contain fluent combinations of words.

A standard language model estimates the probability of a sequence of English words by combining the probability of each word, w_i, in the sequence given the preceding sequence of $i-1$ words. In a similar way, we estimate the probability of a deep syntactic structure d, with root node w_r consisting of l nodes, by combining the probability of each node, w_i, in the structure given the sequence of nodes linked to it via dependency relations that terminates at the node's head. We use the

[4] Morphosyntactic information is left out.

[5] Since we use Factored Models for translating morpho-syntactic information, when computing the translation model we ignore differences in morpho-syntactic information.

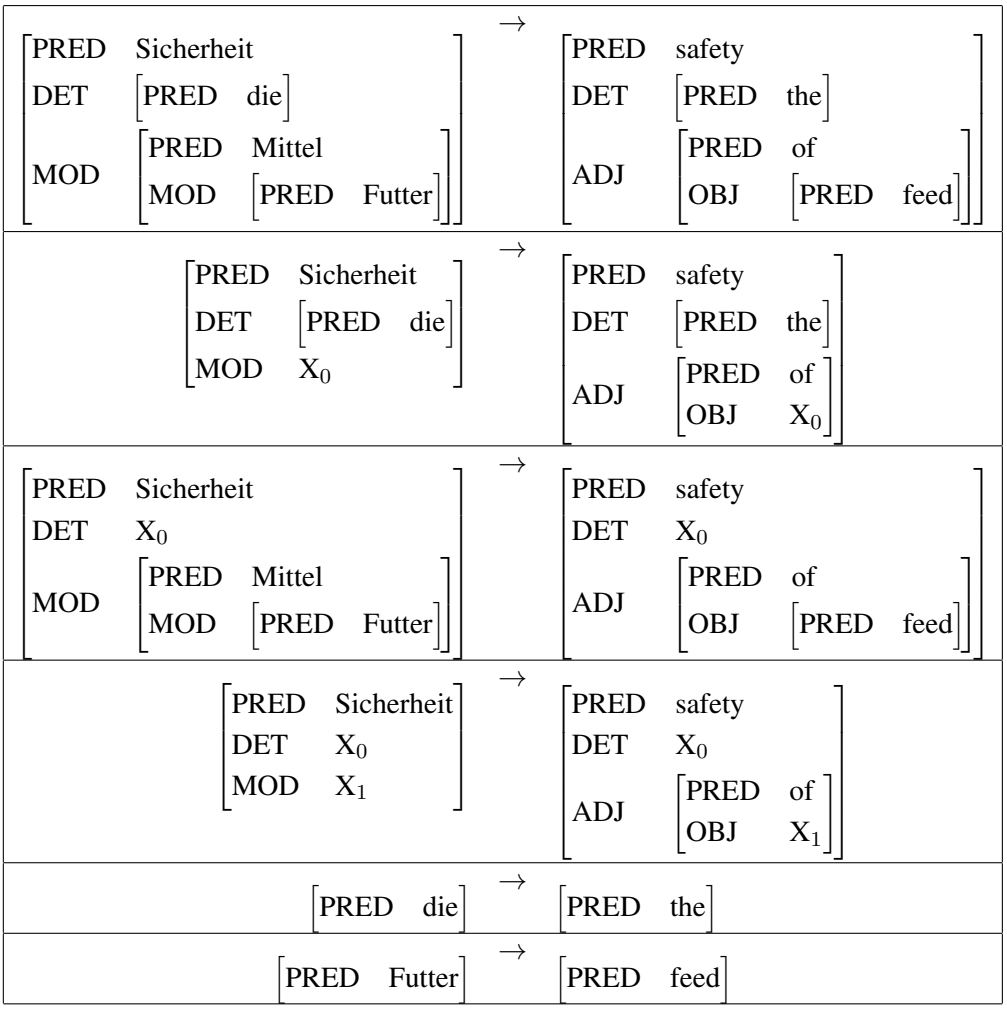


Figure 2: Extracted LFG F-structure transfer rule

function m, to map the index of a node to the index of its head node within the structure.

$$p(d) = \prod_{i=1}^{l} p(w_i | w_r, ..., w_{m(m(i))} w_{m(i)}) \quad (4)$$

In order to combat data sparseness, we apply the Markov assumption, as is done in standard language modeling, and simplify the probability of a deep syntactic structure by only including a limited length of history when estimating the probability of each node in the structure. A trigram deep syntax language model estimates the probability of each node in the structure given the sequence of nodes consisting of *the head of the head of the node* followed by *the head of the node* as follows:

$$p(d) \approx \prod_{i=1}^{l} p(w_i | w_{m(m(i))}, w_{m(i)}) \quad (5)$$

Figures 3(a) and 3(b) show how the trigram deep syntax language model probability is estimated for the English f-structure in Figure 1.[6]

4 Decoding

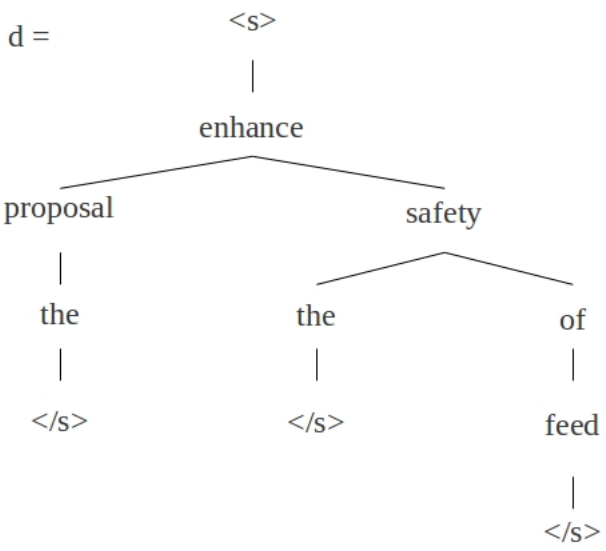
In the (i) parse, (ii) transfer, and (iii) generate architecture of the system, decoding carries out step (ii), the transfer of a SL deep syntactic structure to the target language. Decoding of the SL structure is top-down starting at the root of the structure (usually the main verb of the sentence). Similar to PB-SMT, where decoding search space is exponential in sentence length, our search space is exponential in the number of SL nodes, and we use beam search to manage its size. We use an adaptation of Factored Models (Koehn and Hoang, 2007) to translate morpho-syntactic information.

5 Generation

Generation of the TL output is carried out using XLE rule-based generator (Kaplan et al., 2002), using an English precision grammar (Kaplan et al., 2004; Riezler et al., 2002), designed only to generate fluent sentences of English. When the precision grammar alone is used for generation it often fails due to imperfect input resulting from the transfer step of our system. A fragment grammar is used as a back-off in such cases, to increase the coverage. For some TL structures however, even when the fragment grammar is used the generator can still fail due to ill-formed input structures. The decoder outputs an m-best list of TL structures, the

[6] Argument sharing can occur within deep syntactic structures and in such cases we use a simplification of the actual deep syntax graph structure by introducing the restriction that each node in the structure may only have a single mother node (with the exception of the root node which has no mother node), as this is required for the m function.

(a)



(b)

$$
\begin{aligned}
p(\mathrm{d}) \approx \quad & p(\text{ enhance} \mid \text{<s>}) \\
& p(\text{ proposal} \mid \text{<s> enhance }) \\
& p(\text{ the} \mid \text{enhance proposal }) \\
& p(\text{</s>} \mid \text{proposal the }) \\
& p(\text{ safety} \mid \text{<s> enhance }) \\
& p(\text{ the} \mid \text{enhance safety }) \\
& p(\text{</s>} \mid \text{safety the }) \\
& p(\text{ of} \mid \text{enhance safety }) \\
& p(\text{ feed} \mid \text{safety of }) \\
& p(\text{</s>} \mid \text{of feed })
\end{aligned}
$$

Figure 3: Deep Syntax Language Model Example

content of which tends to vary a lot with respect to lexical choice. By increasing the number of structures input to the generator we can improve overall MT system coverage.

The generator is also non-deterministic, generating a k-best list of output sentences for each input TL structure. For (English) grammatical structures, the value of k is usually low, with the list containing a small number of legitimate variations in word order, and for ungrammatical or ill-formed input structures, k is usually very large, with the lists consisting of many permutations of the same words. Since the transfer decoder outputs an m-best list of structures and for each of those structures we generate k strings, the size of the n-best list for the overall MT system is therefore $m * k$.

Besides increasing coverage, by increasing the value of m, increasing either m or k (or both) also has the potential to reduce search error and result in improved MT system performance. Although the size of m can easily be changed to any desired value for the decoder (by simply changing a value in the configuration file), the generator only allows three options for deterministic versus non-deterministic generation: shortest and longest, generating either only the *shortest* or *longest* sentence with respect to number of words or *allstrings* generating all possible strings given an input structure according to the generation grammar. We refer to the three available generation options as k-options.

In the overall translation model, we include some features that are applied to the TL surface-form sentence after generation.[7] To stay true to the deep syntax approach, we do not use features that use information about the source language surface form word order. We compute a standard language model probability for the generated string and a grammaticality feature function, using information output by the generator about the grammaticality of the string. In addition, we omit scope features from f-structures for rule extraction, transfer and generation.

6 Other Features

In addition to feature functions we described thus far, we include the following additional features:[8]

- lexical translation model for source to target and target to source directions
- transfer rule size penalty (phrase penalty)
- TL node penalty (word penalty)
- fragment penalty

[7] Note that if we did not do this then many the n-best translations would be given the same score, because generation is non-deterministic.

[8] Equivalent features used in PB-SMT are in brackets.

- default transfer rule penalty[9]
- morpho-syntactic rule match feature[10]

7 Evaluation

We provide a detailed evaluation of the system to investigate effects on MT performance of using (i) different methods of word alignment, (ii) restricting the size of transfer rules by imposing different limits on the number of nodes in the LHS and RHS of transfer rules used for transferring SL structures to the TL,[11] (iii) different beam sizes during decoding, (iv) generating different sized m-best TL decoder output structure lists, and (v) different k-options for deterministic versus non-deterministic generation.

German and English Europarl (Koehn, 2005) and Newswire sentences length 5-15 words were parsed using using LFG Grammars (Kaplan et al., 2004; Riezler et al., 2002), resulting in approx. 360K parsed sentences pairs with a disambiguation model used to select the single best parse. A trigram deep syntax language model was trained on the LFG-parsed English side of the Europarl corpus, with approximately 1.26M English f-structures (again using only the single-best parse) by extracting all unigram, bigram and trigrams from the f-structures before running SRILM (Stolcke, 2002). The surface-form language model, used after generation, consisted of the English side of the Europarl, also computed using SRILM. Word alignment was run on the training data yielding an alignment between local f-structures for each f-structure pair in the bilingual training data. All transfer rules consistent with this alignment were extracted. Minimum Error Rate Training (MERT) (Och, 2003) was carried out on 1000 development sentences for each configuration using Z-MERT (Zaidan, 2009).[12]

We restrict our evaluation to short sentences (5-15 words) and use the test set of Koehn et al. (2003), which includes 1755 German-English translations.[13] We carry out automatic evaluation using the standard MT evaluation metric, Bleu (Papineni et al., 2002), in addition to a method of evaluation used to evaluate LFG

[9]When a SL word is outside the coverage of the transfer rules, it gets translated using a default rule that translates any SL word as itself (Riezler and Maxwell, 2006).

[10]For high coverage of transfer rules we allow a fuzzy match between morpho-syntactic information in the SL input structure and those of transfer rules. This feature allows the system to prefer translations constructed from transfer rules that matched the SL structure for a higher number of morpho-syntactic factors.

[11]For example, if the limit is 2, only rules with a maximum of 2 nodes in the LHS and a maximum of 2 nodes in the RHS are used for transfer.

[12]Settings for MERT training were as follows: beam=20, m=100, k=1, k-option=shortest. MERT was carried out separately for each method of word alignment. In all other experiments weights for the DS-INT configuration were used.

[13]The test set was selected on the basis that it is a commonly available test set of short sentences of German to English. Another option would have been to use short sentences from one of the WMT test sets. However, the WMT test sets only contain a relatively low number of short sentences, so instead we revert to the 2003 test set, though a little outdated, is the current best option available.

	Align. Pts.		Rules					
	Total	Ave.	Total	Ave.	Bleu	Prec.	Rec.	F sc.
SF-GDF	4.5M	12.5	2.9M	8.1	1.61	15.83	5.46	8.12
DS-GDF	4.1M	11.5	9.7M	27.1	6.04	29.13	28.17	28.64
DS-INT	2.5M	6.9	13.9M	38.8	16.18	40.31	41.25	40.78

Table 1: Effects of using different methods of word alignment. Note: rule size limit = none, beam = 100, m = 100, k = 1, k-option = shortest

parsers comparing parser-produced f-structures against gold-standard f-structures. The method extracts triples that encode labeled dependency relations, such as *subject(enhance,proposal)* and *object(enhance,safety)* for example, and triples encoding morpho-syntactic information, for example *case(proposal,nominative)* or *tense(enhance,future)*, from each parser produced f-structure and corresponding gold-standard f-structure, counting matching triples to finally compute a single precision, recall and f-score computed over the triples of the entire test set.

We evaluate the highest ranking TL decoder output f-structure with an adaptation of this method since we do not have access to gold-standard f-structures for the test set. Instead we use the next best thing, the parsed reference translations. This provides an evaluation that eliminates generator performance. Note, however, that this method of evaluation is somewhat harsh when used for the purpose of MT evaluation. Since it was designed to evaluate parser output, it does not take differences in lexical choice into account, so, for example, if the MT system produces the correct tense but a different lexical item for *enhance*, such as *tense(improve,future)*, the triple is counted as incorrect ignoring the fact that tense was in fact correct. Correct triples, in the evaluation, are those where the correct lexical choice was made by the system *and* the correct dependency relation (or morpho-syntactic information) was produced.

7.1 Results

Table 1 shows statistics and results for each word alignment method. The deep syntax intersection method of word alignment by far achieves the best result with a Bleu score of 16.18. Results drop sharply when the grow-diag-final algorithm is applied to deep syntax word alignment, with scores of 6.04 Bleu. The method of word alignment that uses the surface-form bitext corpus for word alignment achieves an extremely low score of only 1.61 Bleu.

Table 2 shows automatic evaluation results when different limits on rule size are imposed (all for the best performing alignment method DS-INT). As the limit is increased from 1 node per LHS and RHS to 7 nodes, so does the Bleu score, from 10.09 to 16.55, with a slight decrease, to 16.18, when no limit is put on the size of transfer rules. The biggest increase is seen when we compare the results when the limit is increased from 1 node (10.09 Bleu) to 2 nodes (14.94 Bleu), an increase of almost 5 percentage points absolute. In general, precision, recall and

Limit	Bleu	Prec.	Recall	F-score
1	10.09	38.67	33.89	36.12
2	14.94	41.55	39.09	40.28
3	15.85	41.50	39.93	40.70
4	16.31	41.03	40.25	40.63
5	16.14	40.75	40.50	40.62
6	15.52	40.31	40.71	40.51
7	16.55	40.46	41.03	40.74
none	16.18	40.31	41.25	40.78

Table 2: Effects of limiting transfer rule size. Note: word alignment = DS-INT, beam = 100, m = 100, k = 1, k-option = shortest

f-score also increase, as we increase the limit on transfer rule size, for example, from an f-score of 36.12 when the limit is 1 to 40.74 for a limit of 7.

Results for the system for different decoder beam sizes are shown in Table 3.[14] Results show that changing the beam size does not have a dramatic effect on the system performance. However, the difference between the highest and lowest scores is approximately half a Bleu point, which is a notable decrease in translation quality when the beam is increased from size 10 to 400. This is counter to our expectations, since with an increase in beam size we expect to observe an improvement in Bleu score since more target language f-structures are reached by the decoder search. This indicates that the model used to rank target language solutions is introducing error as some target language f-structures reached when the beam size is 400 are incorrectly ranked higher than other solutions reached when the beam size is 10. In addition, due to the extensive resources and time required to carry out minimum error rate training for the system, the same weights were used for all beam sizes (via optimization with a beam size of 100), and the particular weights may by chance be more suited to solutions reached by a beam size of 10. Further investigation is required before we can make any more general statement about what beam size might be best for f-structure transfer.

Table 4 shows automatic evaluation results for different m-best list sizes.[15] Results show that increasing the size of the m-best list of TL structures produced by the decoder, has a dramatic effect on system performance, with the largest increase in results when we increase the size of m from 1 (12.67 Bleu) to 10 (15.34 Bleu), an increase of almost 3 Bleu points absolute. Results increase again when we increase m to 100 (16.18 Bleu) and again for 1000 (16.57). We include Bleu scores for when true casing is used, and, as expected, for all configurations the Bleu score

[14]Note in this experiment that results are lower relative to other experiments because m=1, as when m is larger than the specified beam size, the decoder can increase the beam size in order to ensure enough solutions.

[15]Precision, recall and f-scores are the same for each configuration, since scores are computed on the highest ranking TL structure, which is the same in each configuration. Bleu-tc scores are for Bleu evaluation with true casing.

Beam	Bleu	Prec.	Recall	F-score
1	12.76	40.61	41.19	40.90
5	12.84	40.70	41.54	41.11
10	13.03	40.79	41.43	41.11
20	12.83	40.69	41.31	41.00
50	12.69	40.35	41.18	41.00
100	12.67	40.31	41.25	40.78
200	12.67	40.24	40.99	40.61
400	12.52	40.06	40.78	40.78

Table 3: Effects of increasing the decoder beam size. Note: word alignment = DS-INT, rule size limit = none, *m* = 1, k = 1, k-option = shortest

m-best list size	Bleu
1	12.67
10	15.24
100	16.18
1000	16.57

Table 4: Effect of increasing the size of the m-best decoder output lists. Note: word alignment = DS-INT, rule size limit = none, beam = 100, k = 1, k-option = shortest. Precision = 40.31%, recall = 41.25%, f-score = 40.78%

drops when casing is taken into account, by approximately 1 Bleu point absolute.

Table 5 shows automatic evaluation results for different generation configurations.[16] The lowest result is seen for deterministic generation with k-option *longest* (15.55), where the generator outputs the longest result, while selecting the shortest generator output string for each TL structure results in an increase to 16.18 Bleu, an increase of almost 1 Bleu point. When non-deterministic generation is used and the generator produces all TL strings for the TL input structure the score increases again to 17.29 Bleu.

[16]Precision, recall and f-scores are the same for each method, since scores are computed on the highest ranking TL structure before generation is carried out.

k-option list size	Bleu
longest	15.55
shortest	16.18
allstrings	17.29

Table 5: Deterministic versus non-deterministic generation. Note: word alignment = DS-INT, rule size limit = none, beam = 100, m = 100. Precision = 40.31, recall = 41.25 and f-score = 40.78 for three configurations.

7.2 Discussion

In the sections that follow, we provide some discussion of results observed.

7.2.1 Word Alignment

Results show that system performance varies dramatically depending on how word alignment is carried out and this is caused by each word alignment method producing different quality alignment points and constraining transfer rule extraction differently (Table 1). The best performing method, DS-INT, produces the fewest and highest quality alignment points and subsequently the best MT performance.

7.2.2 Limiting Transfer Rule Size

In general, as we increase the limit on transfer rule size (Table 2), results improve as more fluent combinations of words in TL structures are produced. Larger snippets of TL structure are also less likely to cause clashes with generation constraints. The minor decrease observed when we change from a limit of 7 to no limit on transfer rule size is probably due to a small number of erroneous transfer rules being eliminated when transfer rule size is limited.

7.2.3 Decoder Beam Size

Increasing the beam size of the heuristic search does not dramatically increase MT system performance (Table 3), with a beam size of 10 being sufficient and this is probably due to the search being highly focused on lexical choice, as it is carried out on lemmatized dependency structures with the translation of morpho-syntactic information carried out independently of decoding, using an adaptation of Factored Models.

7.2.4 M-best Decoder Output

Increasing the number of structures generated (Table 4) has a more dramatic effect. When m is increased from 1 to 10, an increase of almost 3 Bleu points absolute is observed and scores increase again when we move to 100 structures by almost 1 Bleu point. Increasing the size of m to 1000 results in an additional increase of 0.39 Bleu points absolute, but a trade-off exists as the increase in computation time required for generation by increasing m from 100 to 1000 is significant, from approximately 2.33 to 26.75 cpu minutes per test sentence.

7.2.5 Deterministic vs. Non-deterministic Generation

Allowing non-deterministic generation (Table 5) results in a significant increase in Bleu score. With respect to the trade-off in additional computation time required by non-deterministic generation, non-deterministic generation indeed is worthwhile,

since the average time for generation is only increased by half a cpu minute per test sentence, from 2.33 (shortest) to 2.83 (allstrings) cpu minutes.

8 Summary

A detailed evaluation of a German-to-English SMT via deep syntactic transfer system was presented in which values of core parameters were varied to investigate effects on MT output. Experimental results show that the deep syntax intersection word alignment method achieves by far the best results for the system, with larger rule size limits also improving translation quality as estimated by Bleu. Varying the beam size does not show dramatic effects on MT performance, with a beam size of only 10 being sufficient for the transfer-based system. In addition, significant gains can be made by increasing the size of the m-best decoder output list to 100 and non-deterministic generation, however with the significant trade-off in overall translation time introduced by generating from multiple target language structures. In future work, we would like to investigate to what degree the same effects are observed when the language direction is changed to English-to-German. Translation into German would be interesting for this approach, since German has more free word order and richer morphology compared to English. However, significant adaptation of existing generation technologies for German would be required before this is possible, since generation from imperfect German f-structures is required.

References

Bojar, Ondřej and Hajič, Jan. 2008. Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation at the 46th Annual Meeting of the Association for Computational Linguistics*, pages 143–146, Columbus, OH.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.

Dalrymple, Mary. 2001. *Lexical-Functional Grammar*. Academic Press.

Ding, Yuan and Palmer, Martha. 2006. Better Learning and Decoding for Syntax Based SMT Using PSDIG. In *Proceedings of the Association for Machine Translation in the Americas Conference 2006*.

Kaplan, Ronald M. and Bresnan, Joan. 1982. Lexical Functional Grammar, a Formal System for Grammatical Representation. In Joan Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pages 173–281.

Kaplan, Ronald M., King, Tracy Holloway and Maxwell, John T. 2002. Adapting existing grammars: the XLE experience. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Kaplan, Ronald M., Riezler, Stefan, King, Tracy Holloway, Maxwell, John T. and Vasserman, Alexander. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Meeting*, pages 97–104, Boston, MA.

Koehn, Philip, Och, Franz Josef and Marcu, Daniel. 2003. Statistical Phrase-based Translation. In *Proceedings of Human Language Technology - North American Chapter of the Association for Computational Linguistics Conference*, pages 48–54, Alberta, Canada.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*, Phuket, Thailand.

Koehn, Philipp and Hoang, Hieu. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republic.

Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondrej, Constantin, Alexandra and Hoang, Evan HerbstHieu. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic.

Och, Franz Josef. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Och, Franz Josef, Tillmann, Christoph and Ney, Hermann. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, College Park, MD.

Papineni, Kishore, Roukos, Salim, Ward, Todd and Zhu, Wei-Jing. 2002. A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Riezler, Stefan, King, Tracy H., Kaplan, Ronald M., Crouch, Richard, Maxwell, John T. and Johnson, Mark. 2002. Parsing the Wall Street Journal Using Lexical Functional Grammar and Discriminitive Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 271–278, Philadelphia, PA.

Riezler, Stefan and Maxwell, John. 2006. Grammatical Machine Translation. In *Proceedings of Human Language Technologies and the 44th Annual Meeting of the Assciation for Computational Linguistics*, pages 248–255, New York City, NY.

Sgall, Petr, Hajicova, Eva and Panevova, Jarmilla. 1986. *The Meaning of the Sentence and its Semantic and Pragmatic Aspects*. Dordrecht: Reidel and Prague: Academia.

Stolcke, Andreas. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Zaidan, Omar. 2009. Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems. *The Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for Machine Translation* 91, 79–88.