Agreement Constraints for Statistical Machine Translation into German

Philip Williams and **Philipp Koehn**
School of Informatics
University of Edinburgh
10 Crichton Street
EH8 9AB, UK
p.j.williams-2@sms.ed.ac.uk
pkoehn@inf.ed.ac.uk

Abstract

Languages with rich inflectional morphology pose a difficult challenge for statistical machine translation. To address the problem of morphologically inconsistent output, we add unification-based constraints to the target-side of a string-to-tree model. By integrating constraint evaluation into the decoding process, implausible hypotheses can be penalised or filtered out during search. We use a simple heuristic process to extract agreement constraints for German and test our approach on an English-German system trained on WMT data, achieving a small improvement in translation accuracy as measured by BLEU.

1 Introduction

Historically, most work in statistical machine translation (SMT) has focused on translation into English. Languages with richer inflectional morphologies pose additional challenges for translation and conventional SMT approaches tend to perform poorly when either source or target language has rich morphology (Koehn, 2005).

For complex source inflection, a successful approach has been to cluster inflectional variants into equivalence classes. This removes information that is redundant for translation and can be performed as a preprocessing step for input to a conventional surface form based translation model (Nießen and Ney, 2001; Goldwater and McClosky, 2005; Talbot and Osborne, 2006).

For complex target inflection, Minkov et al. (2007) investigate how post-processing can be used to generate inflection for a system that produces uninflected output. Their approach is successfully applied to English-Arabic and English-Russian systems by Toutanova et al. (2008).

Another promising line of research involves the direct integration of linguistic information into SMT models. Koehn and Hoang (2007) generalise the phrase-based model's representation of the word from a string to a vector, allowing additional features such as part-of-speech and morphology to be associated with, or even to replace, surface forms during search. Luong et al. (2010) decompose words into morphemes and use this extended representation throughout the training, tuning, and testing pipeline.

Departing further from traditional SMT models, the transfer-based systems of Riezler and Maxwell (2006), Bojar and Hajič (2008), and Graham et al. (2009) employ rich feature structure representations for linguistic attributes, but have so far been limited by their dependence on non-stochastic parsers with limited coverage. The Stat-XFER transfer-based framework (Lavie, 2008) is neutral with regard to the rule acquisition method and the author describes a manually developed Hebrew-English transfer grammar, which includes a small number of constraints between agreement features. In Hanneman et al. (2009) the framework is used with a large automatically-extracted grammar, though this does not use feature constraints.

In this paper we propose a model that retains the use of surface forms during decoding whilst also checking linguistic constraints defined over associated feature structures. Specifically, we extend a string-to-tree model by adding unification-based

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 217–226,
Edinburgh, Scotland, UK, July 30–31, 2011. ©2011 Association for Computational Linguistics

constraints to the target-side of the synchronous grammar. We suggest that such a constraint system can:

- improve the model by enforcing inflectional consistency in combinations unseen by the language model
- improve search by allowing the early elimination of morphologically-inconsistent hypotheses

To evaluate the approach, we develop a system for English-German with constraints to enforce intra-NP/PP and subject-verb agreement, and with a simple probabilistic model for NP case.

2 Preliminaries

There is an extensive literature on constraint-based approaches to grammar, employing a rich variety of terminology and linguistic devices. We use only a few of the core ideas, which we briefly describe in this section. We borrow the terminology and notation of PATR-II (Shieber, 1984), a minimal constraint-based formalism that extends context-free grammar.

Central to our model are the concepts of *feature structures* and *unification*. Feature structures are of two kinds:

- *atomic* feature structures are untyped, indivisible values, such as `NP`, `nom`, or `sg`
- *complex* feature structures are partial functions mapping features to values, the values themselves being feature structures.

Complex feature structures are conventionally written as attribute-value matrices. For example, the following might represent lexical entries for the German definite article, *die*, and the German noun, *Katze*, meaning *cat*:

$$\textit{die} \quad \longrightarrow \quad \begin{bmatrix} \textsc{pos} & \texttt{ART} \\ \textsc{agr} & \begin{bmatrix} \textsc{case} & \texttt{acc} \\ \textsc{decl} & \texttt{weak} \\ \textsc{gender} & \texttt{fem} \\ \textsc{number} & \texttt{sg} \end{bmatrix} \end{bmatrix}$$

$$\textit{Katze} \quad \longrightarrow \quad \begin{bmatrix} \textsc{pos} & \texttt{NN} \\ \textsc{agr} & \begin{bmatrix} \textsc{case} & \texttt{acc} \\ \textsc{gender} & \texttt{fem} \\ \textsc{number} & \texttt{sg} \end{bmatrix} \end{bmatrix}$$

An equivalent representation, and the one we use for implementation, is that of a rooted, labelled, directed acyclic graph.

A value belonging to a complex feature structure can be specified using a path notation that describes the chain of features in enclosing feature structures. In the examples above, the path $\langle$ AGR GENDER $\rangle$ specifies the atomic value `fem`.

Informally, *unification* is a merging operation that given two feature structures, yields the minimal feature structure containing all information from both inputs. A unification failure results if the input feature structures have mutually-conflicting values. The subject of unification, both in the context of natural language processing and more generally, is surveyed in Knight (1989). In this work, we use destructive graph-based unification, which results in the source feature structures sharing values upon unification.

For example, the result of unifying the agreement values for the feature structures above would be:

$$\textit{die} \quad \longrightarrow \quad \begin{bmatrix} \textsc{pos} & \texttt{ART} \\ \textsc{agr} & \boxed{1} \begin{bmatrix} \textsc{case} & \texttt{acc} \\ \textsc{decl} & \texttt{weak} \\ \textsc{gender} & \texttt{fem} \\ \textsc{number} & \texttt{sg} \end{bmatrix} \end{bmatrix}$$

$$\textit{Katze} \quad \longrightarrow \quad \begin{bmatrix} \textsc{pos} & \texttt{NN} \\ \textsc{agr} & \boxed{1} \end{bmatrix}$$

The index boxes are used to indicate that a value is shared.

3 Grammar

In this section we describe the synchronous grammar used in our string-to-tree model. Rule extraction is similar to the syntax-augmented model of Zollmann and Venugopal (2006), though we do not use extended categories in this work. We then describe how we extend the grammar with target-side constraints.

3.1 Synchronous Grammar

Our translation model is based on a synchronous context-free grammar (SCFG) learned from a parallel corpus. Rule extraction follows the hierarchical phrase-based algorithm of Chiang (2005; 2007). Source non-terminals are given the undistinguished label X, whereas the target non-terminals are given part-of-speech and constituent labels obtained from a parse of the target-side of the parallel corpus. Rules in which the target span is not covered by a parse tree constituent are discarded.

Compared with the hierarchical phrase-based model, the restriction to constituent target phrases reduces the total grammar size and the addition of linguistic labels reduces the problem of spurious ambiguity. We therefore relax Chiang's (2007) rule filtering in the following ways:

1. Up to seven source-side terminal / non-terminal elements are allowed.
2. Rules with scope greater than three are filtered out (Hopkins and Langmead, 2010).
3. Consecutive source non-terminals are permitted.
4. Single-word lexical phrases are allowed for hierarchical subphrase subtraction.

3.2 Constraint Grammar

We extend the synchronous grammar by adding constraints to the target-side. A constraint is an identity between either:

i) feature structure values belonging to two rule elements,

ii) a feature structure value belonging to a rule element and a constant value, or

iii) a feature structure value belonging to a rule element and a random variable with an associated probability function

For example, the following synchronous rule:

$$\textsc{np-sb} \rightarrow \textit{the}\ \textsc{x}_1\ \textit{cat} \mid \textit{die}\ \textsc{ap}_1\ \textit{Katze}$$

might have the target constraint rule shown in Figure 1.

The first three constraints ensure that any AP has agreement values consistent with the lexical items *die* and *Katze*. The next provides a probability based on the resulting case value. The final two are used to disambiguate between possible parts-of-speech.

Constraints are evaluated by attempting to unify the specified feature structures. A rule element may have more than one associated feature structure, so unification is attempted between all combinations. If no combination can be successfully unified then the constraint fails.

$$\begin{aligned}
&\textsc{np-sb} \rightarrow \textit{die}\ \textsc{ap}\ \textit{Katze}\\
&\quad \langle\ \textsc{np-sb}\ \textsc{agr}\rangle = \langle\ \textit{die}\ \textsc{agr}\rangle\\
&\quad \langle\ \textsc{np-sb}\ \textsc{agr}\rangle = \langle\ \textsc{ap}\ \textsc{agr}\rangle\\
&\quad \langle\ \textsc{np-sb}\ \textsc{agr}\rangle = \langle\ \textit{Katze}\ \textsc{agr}\rangle\\
&\quad \langle\ \textsc{np-sb}\ \textsc{agr}\ \textsc{case}\rangle = C\\
&\quad \langle\ \textit{die}\ \textsc{pos}\rangle = \textsc{art}\\
&\quad \langle\ \textit{Katze}\ \textsc{pos}\rangle = \textsc{nn}
\end{aligned}$$

$$P(C=c) = \begin{cases} 0.990, c = \textsc{nom}\\ 0.005, c = \textsc{dat}\\ 0.004, c = \textsc{gen}\\ 0.001, c = \textsc{acc} \end{cases}$$

Figure 1: Example target constraint rule

Ultimately, all feature structures originate in the *lexicon*, which maps a surface form word to a set of zero or more complex feature structures.

3.3 Some Constraints for German

We now describe the German constraints that we use in this paper. Whilst the constraint model described above is language-independent, the actual form of the constraints will largely be language- and corpus-specific.

In this work, the linguistic annotation is obtained from a statistical parser and a morphological analyser. We use the BitPar parser (Schmid, 2004) trained on the TIGER treebank (Brants et al., 2002) and the Morphisto morphological analyser (Zielinski and Simon, 2009). We find that we can extract useful constraints for German based on a minimal set of simple manually-developed heuristics.

Base NP/PP Agreement

German determiners and adjectives are inflected to agree in gender and number with the nouns that they modify. As in English, a distinction is made between singular and plural number, with most nouns having separate forms for each. Grammatical gender has three values: masculine, feminine, and neuter.

A noun phrase's case is usually determined by its

$$\begin{array}{rcl}
\left\{\begin{array}{l}\texttt{ADJA}, \texttt{ART}, \texttt{NN}, \texttt{PDAT}, \\ \texttt{PIAT}, \texttt{PPOSAT}, \texttt{PWAT}\end{array}\right\} & \rightarrow & \{\texttt{NP}, \texttt{PP}\} \\
\{\texttt{APPR}, \texttt{APPRART}\} & \rightarrow & \{\texttt{PP}\} \\
\{\texttt{ADJA}\} & \rightarrow & \{\texttt{AP}, \texttt{CAP}\} \\
\{\texttt{AP}\} & \rightarrow & \{\texttt{CAP}\} \\
\{\texttt{AP}, \texttt{CAP}\} & \rightarrow & \{\texttt{NP}, \texttt{PP}\}
\end{array}$$

Figure 2: Propagation rules used to capture NP/PP agreement relations

role in the clause. For example, nominative case usually indicates the subject of a verb. The case of a prepositional phrase is usually determined by the choice of preposition.

We model these grammatical properties by i) associating, via the lexicon, a set of possible agreement values with each preposition, determiner, adjective, and noun, and ii) enforcing *agreement relations* through pairwise identities between rule elements (as in the example in Figure 1).

For constraint extraction, we first group parse tree nodes into agreement relations. We use the parse tree labels to determine whether a parent shares agreement information with a child. Figure 2 shows the rules that we used in experiments. These should be read as saying that if a child node has a label that appears on the left-hand side of a rule, r, and its parent node has a label that appears on the right-hand side of r then the parent and child share agreement information.

These rules are applied bottom-up from the preterminal nodes of the training data trees. Agreement relations are merged if they share a common parent. Finally, relations are extended to include child words. Figure 3 shows a sentence pair in which the target-side tree has been annotated to show two NP agreement relations found according to the rules of Figure 2.

Of course, this process is not perfect and finds many spurious relations. We guard against the most frequent errors by:

i) Filtering out relations based on label-patterns found during error analysis (for example, relations containing multiple NN nodes)

ii) Attempting to unify the agreement feature structures of the words and rejecting relations for which this fails

Having annotated the training data trees with agreement relations, rule extraction is extended to accept annotated trees and to generate constraint rules of the form shown in Figure 1. Constraints are produced where any two target-side rule elements belong to a common agreement relation. The resulting constraints are grouped by relation into distinct *constraint sets*.

Subject-Verb Agreement

We add limited subject-verb agreement in a similar manner. The additional propagation rules are given in Figure 4. To determine the subject we rely upon the TIGER treebank's grammatical function labels, which the parser affixes to constituent labels. These are otherwise ignored in all propagation rules.

Probabilistic Constraints for NP Case

We make further use of the treebank's grammatical function labels in order to define probabilistic constraints for noun phrase case. Many of the function labels are strongly biased towards a particular case (NP-TOP uses nominative case in 91.5% of unambiguous occurrences, for example). We estimate probabilities by evaluating NP agreement relations in the training data and counting case-label co-occurrences. Ambiguous case values are ignored. The training data uses only 23 distinct NP labels, most of which occur very frequently, so no smoothing is applied. Table 1 shows the 10 most common labels and their case frequencies.

4 Model

As is standard, we frame the decoding problem as a search for the most probable target language tree $\hat{\mathbf{t}}$ given a source language string $\mathbf{s}$:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} p(\mathbf{t}|\mathbf{s})$$

The function $p(\mathbf{t}|\mathbf{s})$ is modelled by a log-linear sum of weighted feature functions:

$$p(\mathbf{t}|\mathbf{s}) = \frac{1}{Z} \sum_{i=1}^{n} \lambda_i h_i(s, t)$$

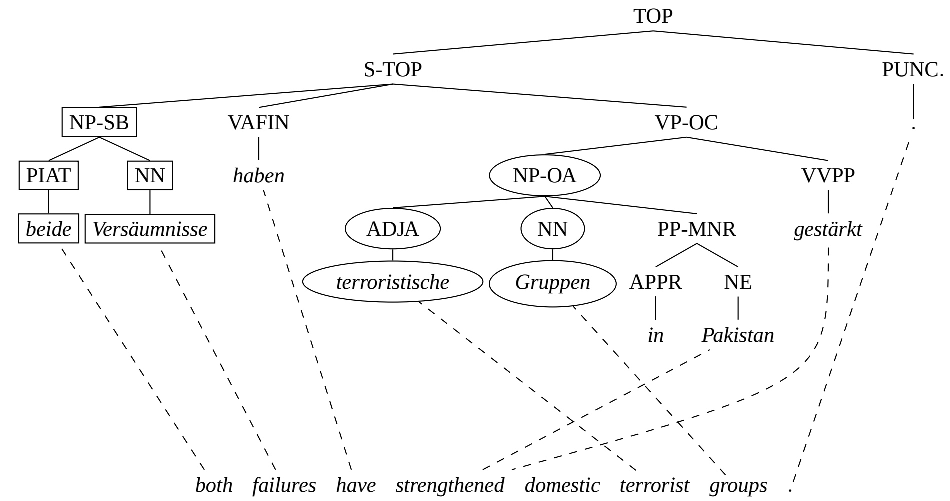

Figure 3: Sentence pair from training data. The two NP agreement relations used for constraint extraction are indicated by the rectangular and elliptical node borders.

$$\begin{array}{rcl} \{\texttt{VAFIN}, \texttt{VMFIN}, \texttt{VVFIN}\} & \rightarrow & \{\texttt{S}\} \\ \{\texttt{NP-SB}\} & \rightarrow & \{\texttt{S}\} \end{array}$$

Figure 4: Propagation rules used to capture subject-verb agreement relations

Label	Nom	Acc	Gen	Dat	Freq
AG	0.1	0.0	99.9	0.0	308156
CJ	10.9	10.3	32.4	46.4	77198
OA	1.6	91.5	0.7	6.2	67686
SB	99.0	0.1	0.4	0.5	60245
DA	1.9	0.2	1.4	96.5	41624
PD	98.2	0.2	1.4	0.3	19736
APP	39.4	7.3	8.7	44.6	7739
MO	18.6	17.3	56.9	7.2	7591
PNC	30.6	0.0	47.4	22.0	4888
OG	0.1	0.0	97.9	2.0	2060

Table 1: The 10 most freqently occurring NP labels with their case frequencies (shown as percentages)

4.1 String-to-Tree Features

Our feature functions include the n-gram language model probability of $\mathtt{t}$'s yield, a count of the words in $\mathtt{t}$'s yield, and various scores for the synchronous derivation. We score grammar rules according to the following functions:

- $p(\mathrm{RHS}_s|\mathrm{RHS}_t, \mathrm{LHS})$, the noisy-channel translation probability.

- $p(\mathrm{RHS}_t|\mathrm{RHS}_s, \mathrm{LHS})$, the direct translation probability, which we further condition on the root label of the target tree fragment.

- $p_{lex}(\mathrm{RHS}_t|\mathrm{RHS}_s)$ and $p_{lex}(\mathrm{RHS}_s|\mathrm{RHS}_t)$, the direct and indirect lexical weights (Koehn et al., 2003).

- $p_{pcfg}(\mathrm{FRAG}_t)$, the monolingual PCFG probability of the tree fragment from which the rule was extracted. This is defined as $\prod_{i=1}^{n} p(r_i)$, where $r_1 \ldots r_n$ are the constituent CFG rules of the fragment. The PCFG parameters are estimated from the parse of the target-side training data. All lexical rules are given the probability 1. This is similar to the p_{cfg} feature used in Marcu et al. (2006) and is intended to encourage the production of syntactically well-formed derivations.

- $exp(1)$, a rule penalty.

4.2 Constraint Model Features

In addition to the string-to-tree features, we add two features related to constraint evaluation:

- $exp(f)$, where f is the derivation's constraint set failure count. This serves as a penalty feature in a soft constraint variant of the model: for each constraint set in which a unification failure occurs, this count is increased and an empty feature structure is produced, permitting decoding to continue.

- $\prod_n p_{case}(c_n)$, the product of the derivation's case model probabilities. Where the case value is ambiguous we take the highest possible probability.

5 Decoding

We use the Moses (Koehn et al., 2007) decoder, a bottom-up synchronous parser that implements the CYK+ algorithm (Chappelier and Rajman, 1998) with cube pruning (Chiang, 2007).

The constraint model requires some changes to decoding, which we briefly describe here:

5.1 Hypothesis State

Bottom-up constraint evaluation requires a feature structure set for every rule element that participates in a constraint. For lexical rule elements these are obtained from the lexicon. For non-lexical rule elements these are obtained from predecessor hypotheses. After constraint evaluation, each hypothesis therefore stores the resulting, possibly empty, set of feature structures corresponding to its root rule element.

Hypothesis recombination must take these feature structure states into account. We take the simplest approach of requiring sets to be equal for recombination.

5.2 Cube Pruning

At each chart cell, the decoder determines which rules can be applied to the span and which combinations of subspans they can cover (the application contexts). An n-dimensional cube is created for each application context of a rule, where $n-1$ is the rank of the rule. Each cube has one dimension per subspan and one for target-side translation options. Cube pruning begins with these cubes being placed into a priority queue ordered according to the model score of their corner hypotheses.

With the introduction of the constraint model, the cube pruning algorithm must also allow for constraint failure. For the hard constraint model, we make the following modifications:

1. Since the corner hypothesis might fail the constraint check, rule cube ordering is based on the score of the nearest hypothesis to the corner that satisifies its constraints (if any exists). This hypothesis is found by exploring neighbours in order of estimated score (that is, without calculating the full language model score) starting at the corner.

2. When a hypothesis is popped from a cube and its neighbours created, constraint-failing neighbours are added to a 'bad neighbours' queue.

3. If a cube cannot produce a new hypothesis because all of the neighbours fail constraints, it starts exploring neighbours of the bad neighbours.

We place an arbitrary limit of 10 on the number of consecutive constraint-failing hypotheses to consider before discarding the cube.

We anticipate that decoding for a highly inflected target language will result in a less monotonic search space due to the increased formation of inflectionally-inconsistent combinations.

6 Experiments

6.1 Baseline Setup

We trained a baseline system using the English-German Europarl and News Commentary data from the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR[1].

The German-side of the parallel corpus was parsed using the BitPar[2] parser. Where a parse failed the pair was discarded, leaving a total of 1,516,961 sentence pairs. These were aligned using GIZA++

[1] `http://www.statmt.org/wmt10/translation-task.html`

[2] `http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html`

and SCFG rules were extracted as described in section 3.1 using the Moses toolkit. The resulting grammar contained just under 140 million synchronous rules.

We used all of the available monolingual German data to train three 5-gram language models (one each for the Europarl, News Commentary, and News data sets). These were interpolated using weights optimised against the development set and the resulting language model was used in experiments. We used the SRILM toolkit (Stolcke, 2002) with Kneser-Ney smoothing (Chen and Goodman, 1998).

The baseline system's feature weights were tuned on the *news-test2008* dev set (2,051 sentence pairs) using minimum error rate training (Och, 2003).

6.2 Constraint Model Setup

A feature structure lexicon was generated by running the Morphisto[3] morphological analyser over the training vocabulary and then extracting feature values from the output.

The constraint rules were extracted using the agreement relation identification and filtering methods described in section 3.3.

We tested two constraint model systems, one using the rules as hard constraints and the other as soft constraints. The former discarded all hypotheses that failed constraints and used the modified cube pruning search algorithm. The latter allowed constraint failure but used the failure count feature as a penalty. Both systems used the NP case probability feature. The weights for these two features were optimised using MERT (with all baseline weights fixed). The systems were otherwise identical to the baseline.

6.3 Evaluation

The systems were evaluated against constrained versions of the *newstest2009*, *newstest2010*, and *newstest2011* test sets. We used a maximum rule span of 20 tokens for decoding. In order that the input could be covered without the use of glue rules (except for unknown words), we used sentences of 20 or fewer tokens, giving test sets of 1,025, 1,054, and 1,317 sentences, respectively. We evaluated translation quality using case-sensitive BLEU-4 (Papineni et al., 2002) with a single reference.

```
(NP-AG der (ADJA regelmäßigen) (ADJA täglichen) (NN Handel))

(PP-MO nach Angaben der (ADJA örtlichen) (NN Index))

(NP-CJ die (ADJA amerikanischen) (NN Blutbad))

(PP-MNR für die (ADJA asiatischen) (NN Handel))

(TOP (NP-SB der (NN Vorsprung) des (NN razor))
     (VVFIN kämpfen)
     (CNP-OA : (NN MP3-Player) (KON und) (NN Mobiltelefone))
     .)
```

Figure 5: Tree fragments containing the first five constraint failures found on the baseline 1-best output

Table 2 shows the results for the three constrained test tests. The p-values were calculated using paired bootstrap resampling (Koehn, 2004). We suspect that the substantially lower baseline scores on the *newstest2011* test set are largely due to recency effects (since we use 2010 data for training).

To gauge the frequency of agreement violations in the baseline output we matched constraint rules to the 1-best baseline derivations and performed a bottom-up evaluation for each target-side tree. For the three constrained test sets, *newstest2009*, *newstest2010*, and *newstest2011*, we found that 15.5%, 14.4%, and 15.6% of sentences, respectively, contained one or more constraint failures. Figure 5 shows the tree fragments for the first five failures found in *newstest2009*.

In order to explore the interaction of the constraint model with search we then repeated the experiments for varying cube pruning pop limits. Figure 6 shows how the mean test set BLEU score varies against pop limit. Except at very low pop limits, the soft constraint system outperforms the hard constraint system. Together with the high p-values for the hard constraint system, this suggests that, despite filtering, our simple constraint extraction heuristics may be introducing significant numbers of spurious constraints. Alternatively, enforcing the hard constraint may eliminate too many hypotheses that cannot be satisifactorily substituted — constraint-satisfying alternatives frequently differ in more than just inflection. Either way, the soft constraint model is able to overcome some of these deficiencies by permitting some constraint failures in the 1-best output.

[3] http://code.google.com/p/morphisto/

	newstest2009-20		newstest2010-20		newstest2011-20	
Experiment	BLEU	p-value	BLEU	p-value	BLEU	p-value
baseline	15.34	-	15.65	-	12.90	-
hard constraint	15.49	0.164	15.95	0.065	12.87	0.318
soft constraint	15.67	0.006	15.98	0.009	13.11	0.053

Table 2: BLEU scores and p-values for the three test sets

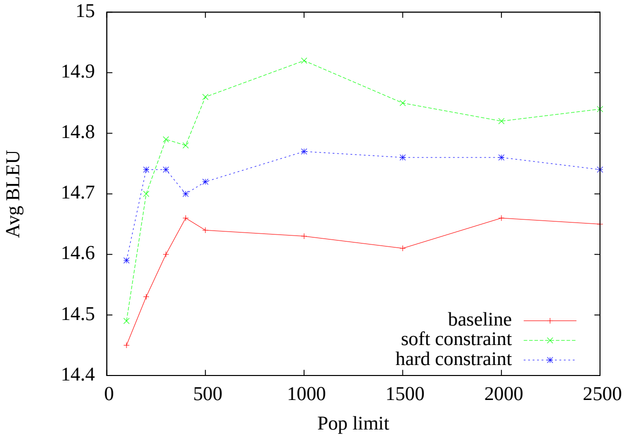

Figure 6: Cube pruning pop limit vs average BLEU score

7 Conclusion

In this paper we have presented an SMT model that allows the addition of linguistic constraints to the target-side of a conventional string-to-tree model. We have developed a simple heuristic method to extract constraints for German and demonstrated the approach on a constrained translation task, achieving a small improvement in translation accuracy.

In future work we intend to investigate the development of constraint models for target languages with more complex inflection. Besides the requirement for suitable language processing tools, this requires the development of reliable language-specific constraint extraction techniques.

We also plan to investigate how the model could be extended to generate inflection during decoding: a complementary constraint system could curb the overgeneration of surface form combinations that has limited previous approaches.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. This work was supported by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme) and made use of the resources provided by the Edinburgh Compute and Data Facility.[4] The ECDF is partially supported by the eDIKT initiative.[5] This work was also supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022. The first author was supported by an EPSRC Studentship.

References

Ondřej Bojar and Jan Hajič. 2008. Phrase-based and deep syntactic english-to-czech statistical machine translation. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 143–146, Morristown, NJ, USA. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.

J.-C. Chappelier and M. Rajman. 1998. A generalized cyk algorithm for parsing stochastic cfg. In *Proceedings of the First Workshop on Tabulation in Parsing and Deduction*, pages 133–137.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. Association for Computational Linguistics.

[4] `http://www.ecdf.ed.ac.uk`
[5] `http://www.edikt.org.uk`

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

Sharon Goldwater and David McClosky. 2005. Improving statistical mt through morphological analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 676–683, Morristown, NJ, USA. Association for Computational Linguistics.

Yvette Graham, Anton Bryl, and Josef van Genabith. 2009. F-structure transfer-based statistical machine translation. In *In Proceedings of Lexical Functional Grammar Conference 2009*.

Greg Hanneman, Vamshi Ambati, Jonathan H. Clark, Alok Parlikar, and Alon Lavie. 2009. An improved statistical transfer system for french–english machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 140–144, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 646–655, Cambridge, MA, October. Association for Computational Linguistics.

Kevin Knight. 1989. Unification: a multidisciplinary survey. *ACM Comput. Surv.*, 21(1):93–124.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *In Proceedings of EMNLP, 2007*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

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

Alon Lavie. 2008. Stat-xfer: a general search-based syntax-driven framework for machine translation. In *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, CICLing'08, pages 362–375, Berlin, Heidelberg. Springer-Verlag.

Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. 2010. A hybrid morpheme-word representation for machine translation of morphologically rich languages. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 148–157, Cambridge, MA, October. Association for Computational Linguistics.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Spmt: statistical machine translation with syntactified target language phrases. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Morristown, NJ, USA. Association for Computational Linguistics.

Einat Minkov, Kristina Toutanova, and Suzuki Hisami. 2007. Generating complex morphology for machine translation. In *Proceedings of the ACL*.

Sonja Nießen and Hermann Ney. 2001. Toward hierarchical models for statistical machine translation of inflected languages. In *Proceedings of the workshop on Data-driven methods in machine translation*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stefan Riezler and John T. Maxwell, III. 2006. Grammatical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 248–255, Morristown, NJ, USA. Association for Computational Linguistics.

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stuart M. Shieber. 1984. The design of a computer language for linguistic information. In *Proceedings of the 10th international conference on Computational linguistics*, COLING '84, pages 362–366, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Intl. Conf. Spoken Language Processing, Denver, Colorado, September 2002*.

David Talbot and Miles Osborne. 2006. Modelling lexical redundancy for machine translation. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 969–976, Morristown, NJ, USA. Association for Computational Linguistics.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *Proceedings of ACL, Association for Computational Linguistics, June 2008*.

Andrea Zielinski and Christian Simon. 2009. Morphisto –an open source morphological analyzer for german. In *Proceeding of the 2009 conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pages 224–231, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *StatMT '06: Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141, Morristown, NJ, USA. Association for Computational Linguistics.