EXPLOITING XLE’s FINITE STATE INTERFACE IN LFG-BASED STATISTICAL MACHINE TRANSLATION

Eleftherios Avramidis and Jonas Kuhn
Linguistics Department, University of Potsdam

Proceedings of the LFG09 Conference

Miriam Butt and Tracy Holloway King (Editors)

2009

CSLI Publications

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

Abstract

We present the addition of a morphological generation component to an LFG-based Statistical Machine Translation System, taking advantage of existing morphological grammars and the FST (Finite State Transducer) processing pipeline of the XLE system. The extended syntax-driven translation system takes separate stochastic decisions for lemmata and morphological tags; the role of finite-state morphological grammars is to generate full forms out of a bundle of morphological tags produced by the translation component. This technique can lead to a more effective use of a given amount of training data from a parallel corpus, since lexical vs. morphosyntactic translation patterns can be induced independently.

The existing FST processing cascade for German, when added to the Statistical Machine Translation System, suffers from generation failures. These occur due to overgeneralisation by the syntax-driven translation process and originate from (i) the use of various underspecification tags in the morphological grammar, or (ii) erroneous assignment of certain tags to a given lemma. In order to deal with this, we add a set of replacement/correction rules on top of the cascade. The augmented FST cascade leads to an increase of generation coverage from 47.90% to 75.35%. A detailed error analysis for the remaining 24.65% is given.

1 Introduction

In current work on Machine Translation (MT), purely data-driven, statistical approaches, based on very large corpora of sample translations, continue to lead to the best evaluation results, at least when tested on the same text domain as they were trained on (Callison-Burch et al., 2008). At the same time, it is conceptually clear that there are limitations to picking up certain generalisations (which can be easily described in linguistic terms) from unstructured training data – Zipf's law has it that the multitude of types of linguistic units occur rather infrequently in corpus data. Hence, an obvious goal for linguistically grounded natural language processing (NLP) research is to find effective combinations of the highly successful statistical techniques with insights from deep linguistic processing. This goal is considerably more challenging than one may first think: nearly all previous experiments on the straightforward ways of constraining the statistical models to apply only on linguistically warranted units have led to a drop in performance (e.g., Koehn et al. (2003a); Chiang (2005)). This is presumably so because the unconstrained system will quite often learn to produce a reasonable translation for some combination of words that does not form a linguistic unit at any level. The development of more structured statistical translation models, capable of incorporating linguistic knowledge while not suffering from a reduced amount of training data, remains a major goal for NLP research for the next years.

[†] The work reported in this paper was supported by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) in the Emmy Noether project PTOLEMAIOS, on Grammar Induction from Parallel Corpora.

In this paper, we focus on the so-called "generation issues" in data-driven translation. These issues arise when translating from a morphologically poor language (e.g. English), into a language that requires complex morphosyntactic rules to be taken into account (e.g. German). The purely statistical MT systems have very limited capabilities of inducing the generalizations behind the morphosyntactic patterns of a language. This occurs as they rely on statistically trained word alignments, which were trained on a parallel corpus. For a given portion of text in the source language, the word-aligned word sequences in the target language are considered as translation candidates. The candidate word sequences are then assembled, mainly on the basis of statistical (n-gram) language models, which assign higher scores to typical word sequence patterns in the target language. Patterns involving high-frequency items will typically be reflected in the language model, but the patterns are not represented systematically and cannot be generalised to lower-frequency items.

The statistical translation approach which we build on (Galley et al., 2004; Hopkins and Kuhn, 2007b) has the ability to separate lexical from morphosyntactic effects during training, since it is driven by a rich syntactic source language analysis (c-structure augmented by features from f-structure). However, it can only produce the specific target language word forms that are included in the training data. This approach suffers to some degree from similar generation issues as the pure statistical MT approach. Given that high-quality morphological analysers exist for many languages, we can see a reasonable extension of this approach: We assume not only a syntactic analysis of the source language, but also a (disambiguated) morphological analysis of the target language. In this paper, we present such an extension building on the English and German resources from the ParGram project (Butt et al., 2002) and focus on the steps needed to ensure robustness of the resulting overall system. Specifically, the cascade of Finite State Transducers is adapted in order to fit the requirements of MT generation.

In the remainder of section 1, a more detailed motivation for morphologically informed generation in data-driven MT is given. In section 2, we first sketch the broader research framework in which our experimental statistical LFG-based translation system is situated, referring to related work and the potential of using LFG in statistical MT. We also present the translation approach that we are building on and show how morphological generation can be integrated straightforwardly in the statistical modelling and combined with standard FSTs. In Section 3 we address issues that arise in the use of specific existing finite state morphological analysers and we describe the adaptation methods employed. Section 4 presents the experiment set-up for an English-to-German translation scenario, evaluation results of system coverage and an error analysis. In section 5, we briefly discuss future directions of our work, before closing with a short conclusion in section 6.

1.1 Motivation: Morphology issues in Statistical MT

LFG is an excellent candidate for exploring sophisticated ways of combining statistics and deep linguistic analysis, thanks to the assumption of parallel correspondence across levels (Riezler and Maxwell, 2006; Hopkins and Kuhn, 2007a). In the case of the present proposal, we follow the LFG-based statistical translation approach of Hopkins and Kuhn (2007a,b), which already exploits c-structure and f-structure information in the source language. We then take advantage of the syntax-morphology interface of an LFG grammar for the target language, which acts as a reliable tool for disambiguating the options that a morphological analyser of the target language produces.[1] This seems to be a useful tool for an exploitation of linguistic generalizations in data-driven MT: when translating from English into a language that is morphologically more challenging (in our case, German), a linguistically motivated morphological analyser can break down word forms into a lemma and a particular set of morphosyntactic features (e.g., {`Mann +NN .Masc .Gen .Sg`} for the form *Mannes* ('man's')).

Lexical generalizations can then be learned for a lemma and generalised to other forms than the ones seen in training. For instance (fig. 1), the system may have learned two things:

(a) *starke Unwetter* (lit. 'strong un-weathers') is a good translation for 'heavy windstorms', and

(b) an *after*-PP in English should be translated as a *nach*-PP in German, where the determiner and attributive adjectives should occur in their dative form (this generalization could have been picked up from examples like 'after a long game' → *nach einem langen Spiel*).

From these two patterns, the system will for instance be able to infer correct translations for the phrases 'a heavy windstorm' as *ein starke**s** Unwetter* and 'after a heavy windstorm' as *nach einem starke**n** Unwetter*, even if the respective form of the adjective did not occur in the respective context in the training data – in fact even if the specific form never occurred in the training data at all.

The benefits to the system can be also interpreted as *a way to make clearer translation decisions*, if we consider the stochastic background of how the words in the two languages are automatically aligned: In the example above, there would be more than 5 candidates for translating the article 'a', which consist of the German indefinite article in various variations of genders and cases. As a simplified example, a pure word-to-word statistical translation model (Brown et al., 1990) would in principle handle this in the same way as a lexical ambiguity; it would create a set of translation candidates and each of these candidates would be assigned

[1] The statistical system uses only the disambiguated morphological analysis of the target language from the training data, not the syntactic analysis itself. In principle, it would be possible to employ a tree-to-tree translation model (Yamada and Knight, 2001; Koehn and Knight, 2003; Huang et al., 2006); however, tree-to-string translation with the capability for morphological generalizations may be a very effective middle ground.

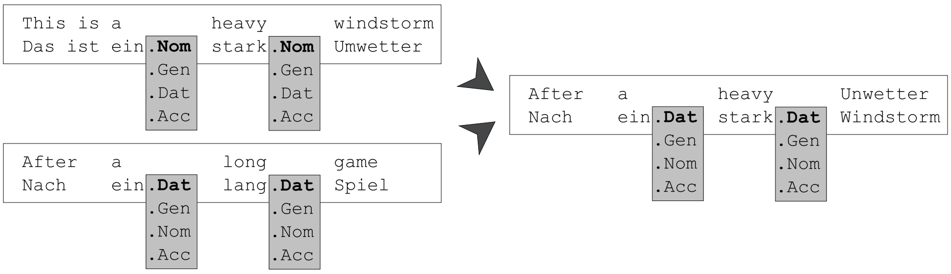
This is a heavy windstorm
Das ist ein .Nom stark .Nom Umwetter
.Gen .Gen
.Dat .Dat
.Acc .Acc
After a long game
Nach ein .Dat lang .Dat Spiel
.Gen .Gen
.Nom .Nom
.Acc .Acc
After a heavy Unwetter
Nach ein .Dat stark .Dat Windstorm
.Gen .Gen
.Nom .Nom
.Acc .Acc

Figure 1: Using the a morpheme generator allows for better coverage for unknown inflections

a probability, depending on how many times it appears in the training corpus as translation of ‘a’ (its *relative frequency*). This appears conceptually weak, as it is obvious that the decision on the correctly inflected form has nothing to do with the relative frequency of this particular form, but is dependent on rules of syntax and agreement in the processed sentence. In our example, if *ein* appears more often, it would have a higher translation probability and therefore would be more likely to be chosen, even if it grammatically should not.

The state-of-the-art systems (following Koehn et al. (2003b)) have reduced this drawback by using multi-word units (the so-called *phrases*) and *language models* which penalise translation sequences which are non-fluent. Nevertheless, the issue can still be complex on the lexical level, as the previously mentioned candidate list may contain both candidates for lexical ambiguities and morphosyntactic inflections. For example, the translation candidates list for the English word `bank` would contain all noun case variations {`Bank` (*nominative/genitive* - prob. 40%), `Flussufer` (*nominative* - prob. 35%), `Flussufers` (*genitive* - prob. 25%)}, but the probability of the most frequent lexical decision is split into two separate hypotheses. When applying the suggested idea, by adding a separate morphology layer, we would reduce this list to {`Bank` (prob. 30%), `Flussufer` (prob. 60%)}, and consequently make the decision on the noun case at a separate stage, with the possibility of considering syntax information, provided from a separate layer of the LFG analysis.

2 Building a morphologically informed system

2.1 Existing work

The idea of augmenting the generation process, when translating into morphologically complex languages, has already been applied to purely statistical systems. Koehn and Hoang (2007), Toutanova et al. (2008) prefer to translate on lemmata and consequently train a separate generation process by using morphological and syntactic factors/features. Other approachess include the use of information from the source-side syntax, aiming to improve the morphology issues at the target side (Minkov et al., 2007; Avramidis and Koehn, 2008). Much research has also been motivated by the needs of agglutinative languages, such as Turkish (El-Kahlout and Oflazer, 2006; Oflazer, 2008), where integrating morphotactic knowledge at the generation stage appears to be essential for creating a fluent output.

2.2 The "tree labelling" approach

For reasons explained in section 1, we attempt to approach the issue in a more linguistically motivated approach, where LFG is the structural backbone of the translation process. We build on top of a statistical tree-to-string translation approach as in Hopkins and Kuhn (2007b), the "PTOLEMAIOS approach", working with the XLE system and the grammars developed in the ParGram project (Butt et al., 2002) and a parallel corpus, word-aligned with GIZA++ (Och and Ney, 2003). Information from the source language LFG analysis drives a "tree labelling" approach to translation: a cascade of statistical (discriminative) classifiers is trained, that traverses the c-structure analysis, taking into account f-structure information and all previous decisions. The new "labels" assigned to the source c-structure tree will contain target language word forms and tree re-structuring instructions (which can have the effect of changing the word order), so a particular target language string can be read off the final tree.

The training process is characterised by the following steps:

(1) Get the XLE parse of the source sentence (e.g. English), add indices for accessing the f-structure information.

(2) For every leaf node, get the corresponding target word from the word alignment with the target (e.g. German) sentence.

(3) Based on the graph structure of the resulting tree/word alignment structure, it is possible to determine a set of "frontier nodes" among the non-terminal nodes, following Galley et al. (2004). The tree/word alignment sub-graphs rooted by these frontier nodes can be used as the building blocks for syntactically informed statistical translation.

(4) Traverse the c-structure tree top-down. In training, we simulate a decision process that subsequently assigns various labels to each tree node. The labels reflect the information needed to reconstruct the full tree/word alignment structure, given only the original source language analysis and the result of previous decisions (e.g., on the mother node). Complex decisions are broken up into simple partial decisions, reflected by sub-labels on the node (For example: Should the node be in the frontier set? Should there be discontinuous parts in the resulting target string? What is the target language word that should be used as a translation for *cooperation*? Should the translation of the right-most daughter precede the translation of the daughter previously translated? etc.).

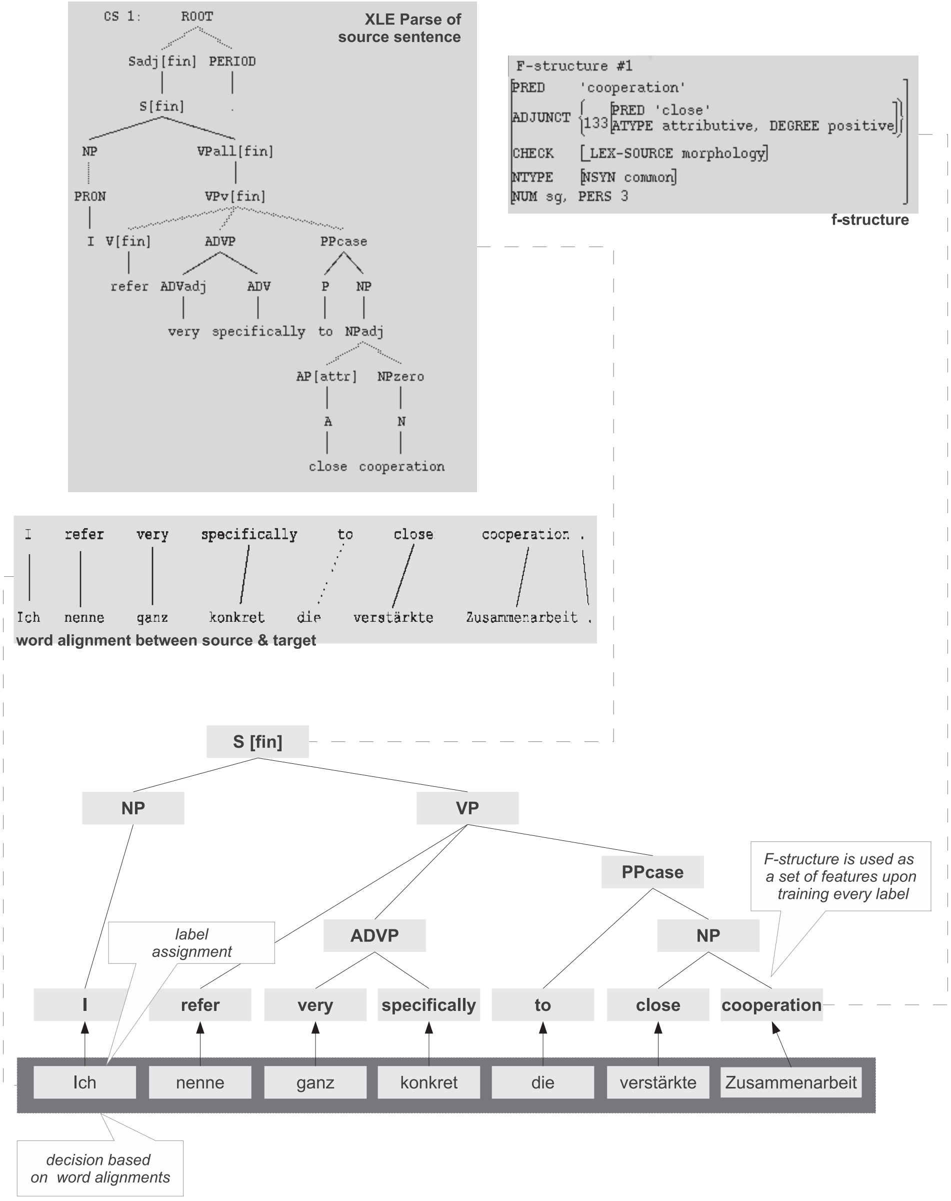
CS 1: ROOT
XLE Parse of source sentence
Sadj[fin] PERIOD
S[fin]
NP
VPall[fin]
PRON
VPv[fin]
I V[fin]
ADVP
PPcase
refer ADVadj
ADV
P NP
very specifically to NPadj
AP[attr] NPzero
A
N
close cooperation
F-structure #1
PRED 'cooperation'
ADJUNCT {133[PRED 'close' ATYPE attributive, DEGREE positive]}
CHECK [LEX-SOURCE morphology]
NTYPE [NSYN common]
NUM sg, PERS 3
f-structure
I refer very specifically to close cooperation .
Ich nenne ganz konkret die verstärkte Zusammenarbeit .
word alignment between source & target
S [fin]
NP
VP
PPcase
F-structure is used as a set of features upon training every label
label assignment
ADVP
NP
I
refer
very
specifically
to
close
cooperation
Ich
nenne
ganz
konkret
die
verstärkte
Zusammenarbeit
decision based on word alignments

Figure 2: Basic training and decoding process

Each sub-step can be characterised as a discriminative classification decision, for which the training data include all the learning features and the correct outcome (label). The learning feature/label combinations for all sub-decisions are collected for the entire training corpus.

(5) The learning feature/label combinations are used for training a (large) set of specialised statistical classifiers that are able to generalise over similar situa-

tions in the top-down tree traversal process. (This involves a very sophisticated back-off technique to ensure that each classifier is based on a sufficiently large sample of evidence.) The resulting cascade of statistical classifiers represents a full tree-to-string translation model.[2]

In the decoding process, i.e., when the model is applied in order to translate a given source language sentence, there is obviously no target language string in the input. This means that step (1) of the training procedure is performed; steps (2) and (3) cannot be performed. This means that the top-down tree traversal of step (4) is performed as a real, cascaded labelling decision process on the nodes (not just as a simulation as in training).[3] The resulting node labels can be used to determine the set of target language word (predicted by the translation model). Then, labels referring to the relative order of the graph fragments indicate the predicted word order.

2.3 Adding the morphology interface

It is conceptually rather simple to augment the cascade of statistical classifiers just described in order to include further labelling decisions. This makes it very straightforward to move away from generation of full word form strings on the target side. Instead, this can be replaced by a more flexible step-by-step generation of lemma information and morphological tags, as they can be used by a finite-state morphological generator. Rather than using full word forms in the tree labels, the first step is just to generate a lemma. The morphological tag specification is then added in separate classification steps, so it can take all available information into account; this information may contain agreement information, based on the analysis done on previously generated words, but may also take advantage of the syntactic analysis of the source sentence, e.g. in order to assign the proper case to the direct and indirect objects.

Whereas the original PTOLEMAIOS approach applies a ParGram LFG grammar to the source language (in our case English) in order to perform the tree labeling, we also parse the target language (German). Since XLE incorporates finite-state transducers (FSTs) for preprocessing (tokenisation) and morphological analysis, the German parses contain a syntactically disambiguated morphological analysis for all words. This is exactly what is needed as training material for the extended tree labelling approach we just described: instead of full form like *starke Unwetter*, we use the following representation of the target language words to train the tree labeler: {`stark +ADJ .Pos .MFNOnly .NA .Pl .St`} {`Unwetter +NN .Neut .NGA .Pl`}. Here, these morphemes are syntactically disambiguated, in the sense that, even if another morphological analysis of

[2] Note that the architecture is not based on the noisy channel model, so in its purest form, the model should not be used in combination with a language model for the target language.

[3] The search strategy adopted is to (greedily) go for the most probable classification outcome in each sub-decision, although in principle it would be possible to use other strategies.

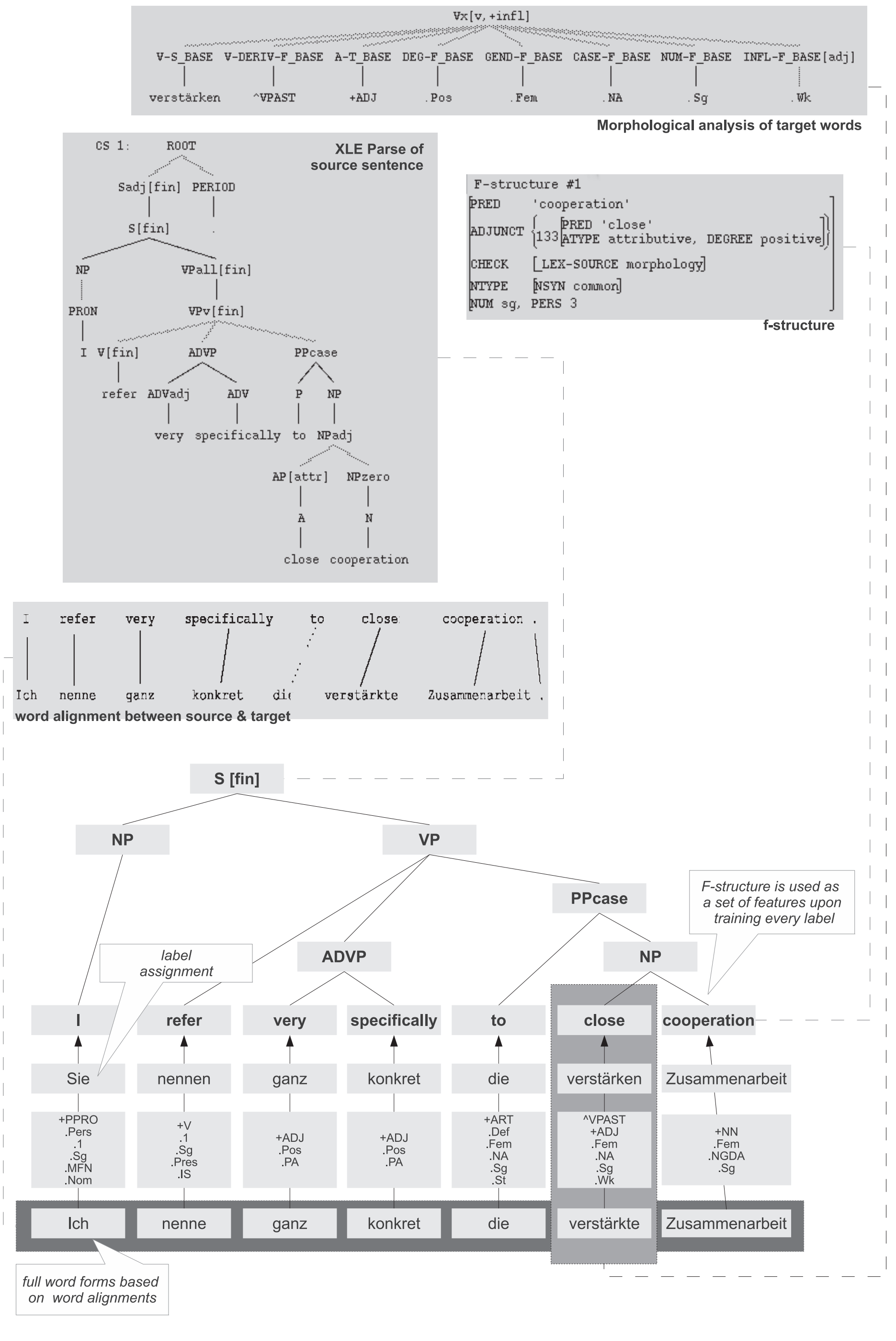
Vx[v,+infl]
V-S_BASE V-DERIV-F_BASE A-T_BASE DEG-F_BASE GEND-F_BASE CASE-F_BASE NUM-F_BASE INFL-F_BASE[adj]
verstärken ^VPAST +ADJ .Pos .Fem .NA .Sg .Wk
Morphological analysis of target words
CS 1: ROOT
XLE Parse of source sentence
Sadj[fin] PERIOD
S[fin]
NP VPall[fin]
PRON VPv[fin]
I V[fin] ADVP PPcase
refer ADVadj ADV P NP
very specifically to NPadj
AP[attr] NPzero
A N
close cooperation
F-structure #1
PRED 'cooperation'
ADJUNCT 133 PRED 'close' ATYPE attributive, DEGREE positive
CHECK LEX-SOURCE morphology
NTYPE NSYN common
NUM sg, PERS 3
f-structure
I refer very specifically to close cooperation .
Ich nenne ganz konkret die verstärkte Zusammenarbeit .
word alignment between source & target
S [fin]
NP
VP
PPcase
ADVP
NP
F-structure is used as a set of features upon training every label
label assignment
I refer very specifically to close cooperation
Sie nennen ganz konkret die verstärken Zusammenarbeit
+PPRO .Pers .1 .Sg .MFN .Nom
+V .1 .Sg .Pres .IS
+ADJ .Pos .PA
+ADJ .Pos .PA
+ART .Def .Fem .NA .Sg .St
^VPAST +ADJ .Fem .NA .Sg .Wk
+NN .Fem .NGDA .Sg
Ich nenne ganz konkret die verstärkte Zusammenarbeit
full word forms based on word alignments

Figure 3: Training and decoding process, after adding the separate morphology layers. Note that here, *ich* is stemmed to *Sie*, because *Sie* has been chosen by the authors of the German LFG Grammar as the citation form of the personal pronoun

the current form would be possible in different contexts, we are getting only the combination of morphemes that matches the most probable syntactic parse.[4]

After training is finished, in application/decoding mode, the tree labelling translator is applied to new input (a set of unseen English sentences) as follows: the English LFG parser is used to produce the most probable c-structure tree, again with information from the f-structure attached on its nodes. The cascade of statistical classifiers is then applied to add the translation labels to this tree, which are then read out to produce a string of lemmata and morphological tags. In the plain tree-to-string approach, the process was finished at this point. Now, we have to perform one last step: the string of lemmata and morphological tags is fed into the target language morphological analyser (run in reverse mode, i.e., as a morphological generator, which is straightforward in finite-state technology).[5]

3 Adapting the morphology interface

The previous section showed that in principle, the tree labelling approach can be straightforwardly extended to produce not just a string of word forms, but a sequence of lemmata and morphological tags used as input for standard FST morphological generation. However, a set of issues arises when this approach is used for a specific morphological grammar, like the one for German used as part of the German ParGram LFG grammar (based on the work by Schiller and Steffens (1990)). In this section, we present the issue and our approach to deal with it in a systematic way.

3.1 The compact underspecified feature format

Using a typical general-purpose morphological analyser for morphologically rich languages such as German, in a different application context than it was originally designed for, may quite naturally lead to complications. Specifically, any pipeline that includes some "soft"/machine learning component feeding the analysis level of the morphological grammar may pose systematic problems. Here we observe this type of problem regarding the set-up of the German morphological grammar, but we present a straightforward solution in the subsequent sections.

To understand the issue, it has to be noted that the feature representations used within the morphological analysers (Schiller and Steffens, 1990) rely on a compact underspecified feature format in order to avoid a proliferation of disjunctive analyses for ambiguous word forms. For instance, the form *Mann* ('*man*') can be either nominative, dative, or accusative singular (only the genitive singular differs:

[4] Some morphological tags are *per se* underspecified (since the form is identical for various feature values, e.g., *starke* could be nominative or accusative, hence the tag `.NA` for Case); here, no disambiguation is needed. We will come back to these underspecified tags in the following section.

[5] Note that the syntactic LFG grammar of the target language is not applied in application/decoding mode, since its only function was to provide a disambiguated morphological analysis of the words in the training data.

Mannes). The morphological grammar assigns the following analysis to *Mann*: {`Mann +NN .Masc .NDA .Sg`}. The case tag `.NDA` combines the tags for nominative, dative and accusative in one compact tag. Other singular nouns are case ambiguous for all four cases, e.g., *Frau* ('woman'), which is assigned the case tag `.NGDA`. Similar tag combinations occur for other morphosyntactic features, such as gender, number, and mood.

This compact feature representation leads to the following issue in translation: as the assignment of labels is trained from output of the morphology, the system will of course pick up generalizations that involve combined tags like `.NDA` . It may turn out however that the translator ends up using such a tag with a lemma that has a slightly different inflection paradigm (e.g., producing {`Frau +NN .Fem` **`.NDA`** `.Sg`} instead of {`Frau +NN .Fem` **`.NGDA`** `.Sg`}). Running the incorrect sequence through the morphological generator will result in a failure.

One may argue that we should try to improve the training so the system will learn to only produce "legal" sequences. However, even if this worked, it would unnecessarily reduce the effectiveness of the training with a given amount of data.[6] It seems much more appropriate to take advantage of the available linguistic knowledge about morphological regularities in the form of morphological analysers and use this to fix the issues.

3.2 The "correction" module

It is relatively straightforward to augment the pre-processing FSTs used in XLE with a "correction" module: using the FST composition operation, we can map combined tags like `.NDA` to other, overlapping combined tags like `.NGDA`, operating in two stages. Hence, a new "recombination" FST is defined, by adding a set of replace rules on top of the existing deep morphology FST, without requiring any modification of the latter.

These extra replace rules could be seen as a *preprocessing step* for the queries that are fed to the generator. They were written manually with regard to the particular morpheme/part-of-speech categories that use a compact representation for ambiguous word forms. Accordingly, their aim is to avoid generation failures, dealing with cases when a probabilistically guessed morpheme does not exactly match the compact morpheme tag expected by the compiled morphology FST. Then they should therefore lead to at least one more compact or more generalised tag, containing the one requested, that could end up in a successful generation. In particular, this task is addressed by:

(a) explicating the combined tags towards their component features (e.g., replacing `.NDA` with `.Nom`, `.Dat` or `.Acc`, disjunctively) and then

(b) generalizing these in order to get a disjunction of all the (other) possible tag combinations that may contain them.

[6] In addition, it should be noted that we are seeing the effect of a representational short-hand that was intended for a different application context.

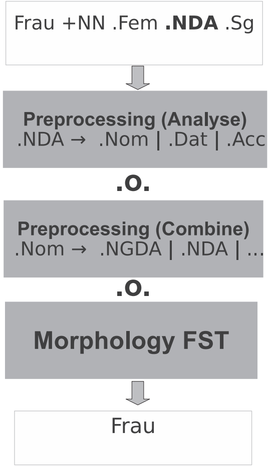
Frau +NN .Fem **.NDA** .Sg
Preprocessing (Analyse)
.NDA → .Nom | .Dat | .Acc
.o.
Preprocessing (Combine)
.Nom → .NGDA | .NDA | ...
.o.
Morphology FST
Frau

Figure 4: Example of a successful generation in an enhanced Finite State Transducers cascade. Whereas *Frau* would previously be generated only if *.NGDA* was given, the replace rules can adapt an input such as *.NDA* so that it is accepted

This way, the desired tags would be taken into consideration, even if they are more or less explicit than the expected. Both stages are compiled out in the resulting recombination FST.

For the German grammar, apart from the noun cases, which have been explained above for means of illustration, rules were written for compact tags referring to verb persons, numbers, genders, moods, and adjective predicate markers.

With the described generalizations, the generator is essentially tuned to overgenerate, in the sense that it will produce all partial tags for a given compact tag (e.g., *nominative, genitive, dative* and *accusative* for `.NGDA`), even if the lemma that the tag is attached to does *not* have the same form for all the feature values. This is intentional since it allows for the desired degree of robustness, i.e., the cascade will typically produce at least one result even for input that would have been incompatible with the original morphological transducer. Since the preprocessing transducers are composed (or cascaded) with the actual morphological grammar transducer, the linguistic knowledge encoded in the latter will constrain the overgeneration. In almost all cases this will have the desired effect, i.e., the correct solution will be included among the solutions. However, it cannot be excluded that an unfortunate combination of overgeneration steps will lead to an incorrect result. What is quite typical is that more than one solution is produced disjunctively. There are ways for the statistical system to choose with some confidence between the alternatives at a later stage (e.g., by scoring the formed phrases with a language model that takes the left and right context in the target language into account).

3.3 Facing incorrect assignments

In the previous section, we addressed cases in which the use of convenient "underspecification tags" in the morphological grammar for ambiguous word forms can lead to issues in the translation-driven construction of the input to a finite-state transducer. One could argue that in the generation of a `.NDA` tag instead of a `.NGDA` tag in translation is not really a mistake, but what we see is a representational issue.

However, in some cases, the step-by-step generation of morphological tags performed in translation may lead to an incorrect assignment of unambiguous feature tags. Since the statistical system has no explicit knowledge of the gender of the nouns, but instead makes predictions based on a wide range of features, it would be possible to assign the tag `.Fem` to a noun that is actually masculine, e.g., leading to {`Mann +NN .Fem .NDA .Sg`}. In such cases, even if most nouns have no flexibility in changing their gender and therefore such a specification in the generation process seems redundant, the nature of the morphology FST would lead it into

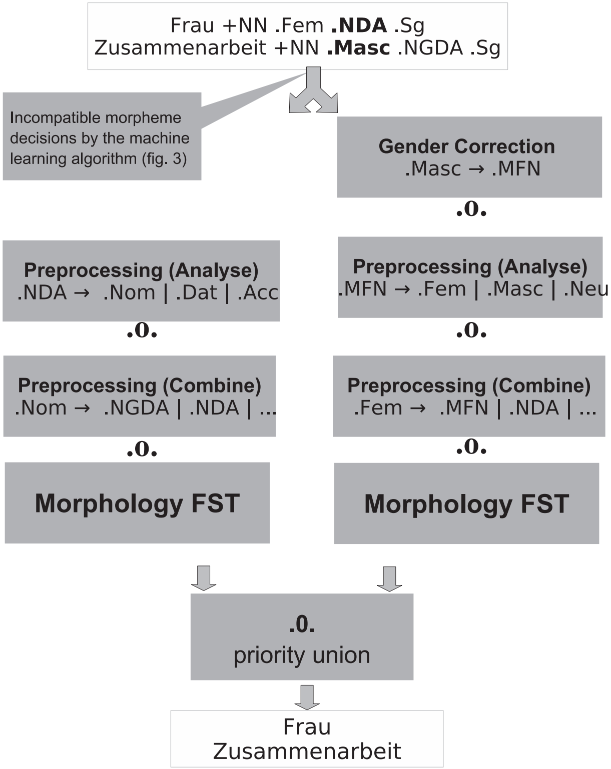

Figure 5: Example of two successful generations in an enhanced Finite State Transducers cascade, extending the one shown at Figure 4. Here, *Zusammenarbeit* (a feminine noun) is generated, although a tag for the masculine form of it has been incorrectly decided by the statistical system.

a generation failure. It is clearly desirable to rely on the morphological grammar for overriding such incorrect feature markings to make the system more robust.

Of course, we only want to change a feature like `.Fem` into `.Masc` in situations where the former analysis is indeed incompatible with the morphological grammar. Adding such a correction to the morphology cascade which was described in the previous section, would correct the issue concerning the nouns, but would cause problems to other parts of speech, for whom the gender information is indeed useful in order to choose within inflection options.

In order to achieve such a flexible manipulation, we actually need a model with two cascades: (a) one with the core of the correction module (section 3.2) without any gender alterations, and (b) an alternative one which takes effect only when the main one fails to generate. This gives all possible gender alternatives as alternative morphemes for the generation failures on the nouns. As mentioned, (b) should only be applied to input (a lemma/tag sequence) for which (a) fails.

3.4 Priority Union

The finite-state operation of *priority union* (Guingne et al., 2003) can be used to this effect (as a unification operation, it was proposed by Kaplan (1995)). By combining two FSTs with priority union (Figure 5), the second FST is only applied to a given input in case the input is not included in the upper side of the first FST. For instance, we may in general apply the mentioned "recombination" FST, and if this combination does not lead to a result, we prefix an additional feature correction FST:

$$\begin{align}
&(T_{recomb} \circ T_{morph}) \overset{p}{\bigcup} (T_{correct} \circ T_{recomb} \circ T_{morph}) \tag{1}\\
&\equiv (T_{recomb} \circ T_{morph}) \cup (\neg upper(T_{recomb} \circ T_{morph}) \circ (T_{correct} \circ T_{recomb} \circ T_{morph})) \tag{2}
\end{align}$$

where T_{morph} is the existing morphology generator, T_{recomb} is the tag recombination transducer and $T_{correct}$ is an FST cascade for substituting tags that may fail during the first generation.

4 Evaluation

The language pair that our experiments focus on is English to German. This pair, in this translation direction, is a good example for disproportional morphology, as German is much more inflected than English. In addition, XLE parsers with the desired morphology features were fully available to us for both languages.

The main focus of our evaluation was how well the morphology interface was adapted to the generation stage of our statistical system. Therefore, we had to measure the improvement in generation coverage. This can be seen as the number of the generations that succeeded, divided by the total number of generations requested. Evaluation of the full translation system will be presented in future work.

System	Generation coverage
only morphology FST	47.90%
compact tags correction	60.41%
gender correction	75.35%

Table 1: Generation coverage, with the various adaptations of the morphology module

4.1 Experiment

The experiment was run on a small, simplified set of the Europarl corpus version 4 (Koehn, 2005). The training set contained 20,000 sentences which had less than 10 words, whereas the untranslated evaluation set contained 1,000 sentences of the same length. The percentages, shown in Table 1, are given based on a proportion over 5510 generation requests.

The results in table 1 show that with the techniques demonstrated, the success rate has been raised from 47.90% to 75.35%, in the full system. The extra correction level for the gender correction by itself was able to improve the coverage by 14.94%, which confirms that the problem was quite critical.

4.2 Error analysis

As the results show, there is still a considerable 24.65% of failures taking place even when all of the above corrections are applied. A first detailed evaluation was performed manually in order to further investigate the actual cause of the failures. It became clear that many failures had common reasons: a more concrete categorisation of approximately 70% of the errors has successfully been traced with regular expressions, whereas the "Wrong POS" category was estimated based on a smaller manually evaluated subset.

The outcome of the analysis is shown in Table 2, in which the percentages sum up to the 24.65% questioned. What is identified as a major cause of failures, includes:

(a) the predefined behaviour of the statistical part of the system, which does not always provide the full set of required morphemes. As this has been the most robust solution, the statistical system first decides the categories of the morphemes that a word may be assigned and then makes a decision for each morpheme value. However, FST allows the morpheme order for a small set of words to vary, especially when these words are generated by combining other smaller words. In this error category we would count *complex prepositions* (like *gegenüber*, *daraus*), prepositions with fused articles (*zur*, *im*), compounds (*Parkordnung* etc.) and some other forms which appear as articles or personal pronouns.

(b) wrong POS behaviour (e.g. when a verb lemma is requested to be inflected as

Error type	failure perc.
fused articles (*zur*, *im*), complex prepositions (*darüber*)	4.33%
wrong POS predicted	3.95%
definite morpheme for indefinite article	2.47%
proper names	2.05%
ihre_attr with incompatible morphemes	1.56%
die_art with incompatible morphemes	1.38%
derivatives of verbs	1.32%
dem with incompatible morphemes	1.16%
compounds	1.03%
NoGend tag required for spec. nouns (*Herr*, *Ausschuss*, *Prozess*)	0.69%
personal pronoun derivatives	0.58%
hyphens	0.18%
numerical expressions	0.18%
Other issues	3.73%

Table 2: Analysis of the persisting generation failures

a noun, or when the system requires an adjective-looking item, which is in fact a derivational form of a verb).

(c) incompatible morphemes (the definite morpheme for indefinite articles, the *.NoGend* morpheme for particular nouns) that should have been included in the correction layer described in section 3.3

(d) proper names, which are not known by the FST. Although the generator reports a failure, they do not consist a translation error, as they can safely be left uninflected.

(e) other issues, such as hyphens, numerical expressions etc.

Many of the issues above could be addressed with some minor machinery alterations. Point (a) above represents a large class of failures. For this case, the statistical system decision process can be adapted in order to deal with morpheme tag sets of variable width. Similarly, incompatible morphemes (b) can be addressed by adding rules as shown in Section 3.3.

5 Future work

Since there is still a small class of generation failures due to various issues (section 4.2), some effort is needed in order to guarantee robustness. We could consider a *backing-off* statistical model, which could perform the tree labeling process in a combined mode, for every sentence: During training, every tree node would get labels referring to both the *full word form* (as in the original system in section 2.2) and the *lemma+tags* (as in the extended system in section 2.3). Then, during the decoding, when the morphology generator fails to produce a word from out of its morphemes, the trained full word form label would be used.

There is also ongoing work in order to efficiently handle the overgenerating phenomenon, which was explained at section 3.2. The method of *n-best re-scoring* (Och and Ney, 2002; Koehn and Knight, 2003), creates a set of alternatives forms of the whole sentence and uses an *n-gram language model* to re-score them, based on their fluency. That could be a useful tool for getting a more certain decision for the outcome of the generations that resulted in several alternative inflections.

Additionally, the order in which the tree nodes are being traversed has an impact on the availability of the agreement features within the sentence. Whereas the experiments were performed on a simple top-down, left-to-right tree traversal (and hence left-right in the sentence), this does not provide enough agreement features from words following the ones we examine at a certain point. For example, the determiners and the adjectives would have more hints for their gender and case, if they know the properties of the following noun. However, nouns normally get traversed and analysed afterwards, since they are to the right of their determiner and adjectival modifiers. We are considering a restructuring on the order of the traversal mechanism, so that there is better availability of such features.

6 Conclusion

We have explained the adaptation of a German Morphology Finite State transducer, so that it can inflect words from given morphemes, as they have been given at the final stage of a LFG-based statistical Machine Translation system. A new “recombination” transducer was formed by writing a set of replace rules on top of the existing morphology transducer. During this adaptation, two major issues were shown to be *(a)* the compact underspecification tags required by the FST, which would not match what was decided by the statistical system and *(b)* the requirements of specific POSs for morphemes that are useful for agreement, but redundant for generation. Both issues, when addressed, led to a significant improvement at the generation coverage.

References

Avramidis, E. and Koehn, P. (2008). Enriching input for statistical machine translation. In *ACL '08: Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics*.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85.

Butt, M., Dyvik, H., King, T., Masuichi, H., and Rohrer, C. (2002). The parallel grammar project. *International Conference On Computational Linguistics*, pages 1–7.

Callison-Burch, C., Koehn, P., Monz, C., Schroeder, J., and Fordyce, C. S., editors (2008). *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Columbus, Ohio.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270.

El-Kahlout, I. D. and Oflazer, K. (2006). Initial explorations in English to Turkish statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 7–14, New York City. Association for Computational Linguistics.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *Proceedings of HLT/NAACL 2004*.

Guingne, F., Nicart, F., Champarnaud, J., Karttunen, L., Gaal, T., and Kempe, A. (2003). Virtual operations on virtual networks: The priority union. *International Journal of Foundations of Computer Science*, 14(6):1055–1070.

Hopkins, M. and Kuhn, J. (2007a). Deep grammars in a tree labeling approach to syntax-based statistical machine translation. In *Proceedings of ACL Workshop on Deep Linguistic Processing 2007*.

Hopkins, M. and Kuhn, J. (2007b). Machine translation as tree labeling. In *Proceedings of NAACL-HLT/AMTA Workshop on Syntax and Structure in Machine Translation 2007*.

Huang, L., Knight, K., and Joshi, A. (2006). Statistical syntax-directed translation with extended domain of locality. *Proc. AMTA*, pages 66–73.

Kaplan, R. M. (1995). Three seductions of computational psycholinguistics. In Dalrymple, M., Kaplan, R. M., Maxwell, J. T., and Zaenen, A., editors, *Formal Issues in Lexical-Functional Grammar*. CSLI Publications.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. *MT Summit*, 5.

Koehn, P. and Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

Koehn, P. and Knight, K. (2003). Feature-rich statistical translation of noun phrases. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003a). Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada.

Koehn, P., Och, F. J., and Marcu, D. (2003b). 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.

Minkov, E., Toutanova, K., and Suzuki, H. (2007). Generating complex morphology for machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the Association of Computational linguistics*, pages 128–135, Prague, Czech Republic. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Oflazer, K. (2008). Statistical machine translation into a morphologically complex language. In *CICLing*, volume 4919 of *Lecture Notes in Computer Science*, pages 376–387. Springer.

Riezler, S. and Maxwell, J. (2006). Grammatical machine translation. In Butt, M., Dalrymple, M., and King, T. H., editors, *Intelligent Linguistic Structures: Variations on themes by Ronald M. Kaplan*, chapter 3, pages 35–52. CSLI Publications, Stanford, CA.

Schiller, A. and Steffens, P. (1990). A two-level environment for morphological descriptions and its application to problems of german inflectional morphology. *Terminology and Knowledge Engineering*, 1:318–329.

Toutanova, K., Suzuki, H., and Ruopp, A. (2008). Applying morphology generation models to machine translation. In *ACL '08: Proceedings of the 46th Annual Meeting of the Association of Computational linguistics*, pages 514–522. Association for Computational Linguistics.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.