# Factor Templates for Factored Machine Translation Models

*Yvette Graham, Josef van Genabith*

National Centre for Language Technology
Dublin City University, Dublin
{ygraham,josef}@computing.dcu.ie

## Abstract

In this paper, we present a method of avoiding the combinatorial explosion encountered in Factored Models during the construction of translation options caused by the large number of possible combinations of target language lemmas and morpho-syntactic factors. We automatically extract *factor templates* from a word-aligned annotated bilingual corpus and use them to distinguish which morpho-syntactic factors should be translated separately from lemmas and in doing so avoid the large number of translation options otherwise considered for generation. Besides Phrase-Based SMT, Factored Models can be applied to SMT via deep syntactic transfer, which is the focus of our work. We therefore include an experimental evaluation of our method for a SMT via deep syntactic transfer system, comparing the baseline standard Factored Model with one that uses *factor templates* for translating morpho-syntactic factors, resulting in a large increase in BLEU score.

## 1. Introduction

Factored Machine Translation Models [1] build on Phrase-Based Models [2] by translating morpho-syntactic information separately from the lemma of source language (SL) words. Factored Models yield richer translation models since they are computed from more general representations of the training data, e.g. lemmas and morphology, when compared with standard Phrase-Based translation models trained on surface-form words. In addition, Factored Models can potentially increase coverage of unseen data, since coverage of inflections of lemmas not seen in bilingual training is possible, as analysis and generation components can be trained on monolingual data.

One particular Machine Translation approach, to which Factored Models is well-suited, is SMT via deep syntactic transfer [3, 4, 5]. In this approach, morpho-syntactic information for source and target words is available for the training data. Unseen source language input is also parsed to the deep syntactic representation, which contains surface form words in lemma form with morpho-syntactic information, which need to be translated to the target language (TL). Richer translation models provided by Factored Models can therefore be computed for the training data and used to translate unseen SL input. Some adaptation is required, however, the most significant of which is that, while Phrase-Based Factored Models can employ word-level generation, Factored Models for SMT via deep syntactic transfer is restricted to sentence-level generation. Sentence-level generation comes with the disadvantage that far more translation options must be pruned prior to TL generation.

Factored Models have already been shown by [4] to improve MT performance for an English to Czech deep syntactic transfer SMT system. However, separating lemmas from morpho-syntactic information, for both Phrase-Based and Deep Syntax Models brings with it some complexity challenges, described in [1] and [4]: the number of translation options generated when combining target language lemmas and morpho-syntactic factors is very large and can become unmanageable. [1] describe how this combinatorial explosion can occur when computing translation options and address the problem by early pruning of expansions, limiting the number of translation options per input phrase to 50, and remark that this solution is not perfect. [4] also report combinatorial explosion as a problem, noting it as a main reason for their system's under-performance, as too many translation options force good solutions off the decoding stacks during the heuristic search.

In this paper, we propose a solution to the combinatorial explosion associated with combining translated lemmas and morphology in Factored Models. We use *factor templates*, extracted from the word-aligned annotated bitext corpus, to avoid the large number of combinations of translated lemmas and morphology. A factor template contains an example set of morpho-syntactic factors from the training data for a phrase pair. A comparison of the SL side of a factor template and the factors belonging to new input data distinguishes factors in the input that need to be translated *separately from the lemma* and factors that can be translated *with the lemma*. Reducing the number of factors that are translated separately from the lemma like this allows us to avoid the large number of combinations of translated lemmas and morpho-syntactic factors otherwise considered for generation.

We provide an experimental evaluation of the method using a deep syntactic transfer SMT system for German to English translation, comparing system performance when all factors are translated separately from the lemma and when factor templates are used to filter which factors are translated separately from input lemmas. We evaluate using the



*Proceedings of the 7th International Workshop on Spoken Language Translation*
*Paris, December 2nd and 3rd, 2010*

standard metric BLEU [6], in addition to a parser evaluation metric that provides a more detailed analysis of individual morpho-syntactic factors using precision and recall measures computed by comparing factors of the MT output with parsed reference translations. Parser coverage restricts our evaluation to short sentences (5-15 words).

## 2. Combinatorial Explosion

The number of translation options for a SL word (or phrase) in Factored Models is $O(e^f)$, where $f$ is the number of SL factors (including the lemma) and $e$ is the number of possible translations for a SL factor. For example, Figure 1 shows the lemma and morpho-syntactic factors for the German word *Mann* and the possible translations into English. The total number of translation options in this simple example is 900.[1] The task of guessing a single correct combination out of this large number of possible combinations is challenging.

### 2.1. Effects of Combinatorial Explosion on Phrase-Based versus Deep Syntactic Transfer Models

The large number of translation options that results from combining translated lemmas and morpho-syntactic factors forces many translation options to be pruned, which is of course a disadvantage because some good translations may never be considered as output. For deep syntactic transfer SMT, a much larger number of translation options must be pruned when compared to Factored Phrase-Based Models, because in deep syntactic transfer, generation is carried out on the sentence level, so the morpho-syntactic factors for the entire sentence must be in place before generation. In addition, generation is relatively slow. Slow sentence level generation results in a very high portion of translation options being pruned.

In addition, in Phrase-Based Factored Models, there are no predefined restrictions on which morpho-syntactic factors are required for translation and generation, which is not the case for deep syntactic transfer, as successful generation relies on the morpho-syntactic factors being in line with rules in the TL generation grammar, so even if a TL morpho-syntactic factor does not seem useful for translation between a specific pair of languages it cannot be left out to reduce the number of translation options, as would be possible in Phrase-Based Factored Models.

## 3. Factor Templates

A factor template can be envisaged as a blue-print for translating a SL phrase into the TL. Each template has a source and target side containing the lemmatized words of an SMT phrase and an example set of morpho-syntactic factors for each source and target lemma. Figure 2(a) shows a factor template for the German-English phrase *neues Haus*|||*new house*.

When translating a SL phrase, the target side of the factor template is used to provide an initial set of translated TL morpho-syntactic factors. The set of factors provided by the template may not contain the correct translation for all of the SL factors, and we use information in the source side of the template to indicate which factors may be incorrect. Figure 2(b) shows how the input SL phrase *neue Häuser* is decomposed into its lemmas, *neu* and *haus* and morpho-syntactic information.

The SL factors of the input words are compared with the SL template factors. Only when a mismatch occurs between a template factor and an input factor, is a factor translated separately from its lemma. All target side factors in the template for which the corresponding SL factor matched that of the the SL input are used as TL output factors. In the example in Figure 2(b), all SL factors in template 2(a) match those of the input except for *number*. Therefore, all TL template factors excluding *number* are used as the TL output factors for *new house*. The value of *number* can then be translated separately from the rest of the translation.

### 3.1. Extracting Factor Templates

Factor templates are automatically extracted from the annotated bitext corpus and only a single template is extracted for each unique lemmatized SMT phrase. For example, both of the following SMT phrases could exist in the phrase table: **(1)** *neues haus ist*|||*new house is*, **(2)** *neue häuser sind*|||*new houses are*, and since both form the same phrase pair when lemmatized, *neu haus sein*|||*new house be*, only a single factor template is extracted along with a single set of morpho-syntactic factors belonging to either one of the SMT phrases.

### 3.2. Avoiding the Combinatorial Explosion

Factor templates reduce the number of factors translated separately from each lemma and in doing so, reduce the overall number of translation options produced when combining translated lemmas and factors. For instance, in the example shown in Figure 2(b), the number of translation options considered for the English phrase with lemmas *new house* is reduced from 1620 to 2 (since *degree* has 3 possible values: *comparative, positive, superlative*; *a-type* 3 possible values: *adverbial, attributive, predicative*; *person* 3 values; *number* 2 values; *case* 2 values; *syn-n-type* 3 values and *common-n-type has 5 possible has values* ).[2]

### 3.3. Idiosyncratic Translations

Factored Models can, in some cases, over-generalize and produce an incorrect translation. Valid idiosyncratic translations exist for words in different language pairs and such exceptional translations can cause problems when the lemma

[1] Morpho-syntactic factors in the example are obtained from Lexical Functional Grammar (LFG) f-structures.

[2] Note that there is an even larger total number of translation options for *neue Häuser*. In the example, we just include translation options for English lemmas *new house*

$$\begin{array}{ccc} \text{Mann} & \rightarrow & \text{man / gentleman / husband / worker / fellow} \\ \begin{bmatrix} \text{pers} & \text{3} \\ \text{num} & \text{singular} \\ \text{case} & \text{dative} \\ \text{gend} & \text{masculine} \\ \text{syn-n-type} & \text{common} \\ \text{common-n-type} & \text{count} \end{bmatrix} & & \begin{bmatrix} \text{num} & \text{singular / plural} \\ \text{pers} & \text{1 / 2 / 3} \\ \text{case} & \text{nominative / oblique} \\ \text{syn-n-type} & \text{common / pronoun / proper} \\ \text{common-n-type} & \text{count / gerund / mass / measure / partitive} \end{bmatrix} \end{array}$$

Figure 1: Example of the combinatorial explosion involved in translating lemmas and morpho-syntactic factors separately.

**(a)** **Factor Template**

$$\begin{array}{ccccc} \text{neu} & \text{haus} & \rightarrow & \text{new} & \text{house} \\ \begin{bmatrix} \text{degree} & \text{positive} \\ \text{a-type} & \text{attributive} \end{bmatrix} & \begin{bmatrix} \text{pers} & \text{3} \\ \text{num} & \text{sg} \\ \text{gend} & \text{neut} \\ \text{case} & \text{nom} \\ \text{syn...} & \text{common} \\ \text{com...} & \text{count} \end{bmatrix} & & \begin{bmatrix} \text{degree} & \text{positive} \\ \text{a-type} & \text{attributive} \end{bmatrix} & \begin{bmatrix} \text{pers} & \text{3} \\ \text{num} & \text{sg} \\ \text{case} & \text{nom} \\ \text{syn...} & \text{common} \\ \text{com...} & \text{count} \end{bmatrix} \end{array}$$

**(b)** **Factored SL Input** **Factored TL Output**

$$\begin{array}{ccccc} \text{neu} & \text{haus} & \rightarrow & \text{new} & \text{house} \\ \begin{bmatrix} \text{degree} & \text{positive} \\ \text{a-type} & \text{attributive} \end{bmatrix} & \begin{bmatrix} \text{pers} & \text{3} \\ \textbf{num} & \textbf{pl} \\ \text{gend} & \text{neut} \\ \text{case} & \text{nom} \\ \text{syn...} & \text{common} \\ \text{com...} & \text{count} \end{bmatrix} & & \begin{bmatrix} \text{degree} & \text{positive} \\ \text{a-type} & \text{attributive} \end{bmatrix} & \begin{bmatrix} \text{pers} & \text{3} \\ \textbf{num} & \textbf{?} \\ \text{case} & \text{nom} \\ \text{syn...} & \text{common} \\ \text{com...} & \text{count} \end{bmatrix} \end{array}$$

Figure 2: **(a)** Factor template for German-English lemmatized phrase pair: *neu haus*|||*new house*, **(b)** Factored SL input phrase for *neue Häuser*, mismatching features in the source input are in bold, and TL output, factors translated separately from the lemma have '?' as a value in the English factored phrase.

is translated separately from its morpho-syntactic information. A classic example is when translating between two languages in which a noun with the same meaning has a different *number* in each language. For example, consider the German phrase *"die Polizei ist"* in which the noun *Polizei* is in the singular and the correct English translation is *"the police are"* in which the translation of *Polizei* is the noun *police* which must be in the plural in English. For Factored Models, if the morpho-syntactic factor *number=singular* of the German lemma *Polizei* is translated into English separately from the lemma *Polizei*, there is a risk of over-generalizing and assigning a high probability to *police, number=singular* in English, which is incorrect. For deep syntactic transfer Factored Models, this over-generalization problem is severe. As mentioned previously, generation operates on the sentence level for deep syntactic transfer, compared to the word level in Phrase-Based Models, and since a large number of translation options are pruned, its very unlikely for a deep syntactic transfer Factored Model to produce the correct translation, *the police, number=plural* if *number* is translated separately from *Polizei*.

Factor templates provide a solution to over-generalizing when translating lemmas separately from their morpho-syntactic information. Figure 3(a) shows a factor template extracted from the corpus for the German-English lemmatized phrase pair *die polizei kommen|||the police come* and Figure 3(b) shows how the template is applied to an input German phrase *die Polizei kommt* analyzed as the (same) lemma sequence, *die polizei kommen*, with morpho-syntactic factors. Since only the factor *tense* mismatches the source side of the template, it is the only factor to be translated separately from the lemma and all other factors, including *number*, are provided by the target side of the factor template. This results in the idiosyncratic translation of *Die Polizei kommt* (where *polizei, number=singular*) being translated correctly into English as *The police are coming* (where *police, number=plural*).

## 4. Translating Mismatching Factors

As described in Section 3, factors in the SL input that mismatch those of the factor template are translated separately from the lemma. For translating mismatching factors, we use a probability distribution computed from the relative frequencies of source and target factors in the word-aligned annotated corpus, $p(v_e|v_f)$, where $v_f$ denotes a SL factor and $v_e$ is a target language factor.

In addition, since we test the method in a deep syntactic transfer SMT system, information about dependency relations between TL words is also available when we translate factors. Intuitively, information about the dependency relations between a word and its head might be useful for translating the morpho-syntactic factor *case*. Therefore, we also compute relative frequencies for *case* conditioning on the dependency relation between a word and its head, $p(v_e|d_e)$, where $d_e$ denotes the dependency relation between the word and its head.

## 5. Evaluation

We evaluate factor templates using a deep syntactic transfer SMT system to translate from German into English. The system uses the Lexical Functional Grammar [7, 8, 9] (LFG) feature structure (f-structure), an attribute-value structure encoding of bilexical labeled dependencies, as the intermediate representation. For the purpose of the evaluation, the system was trained on Europarl [10] and Newswire bilingual corpora parsed with XLE [11] German and English LFG grammars [12, 13].[3] A restricted sentence length of 5-15 words was used for bilingual training resulting in approximately 360K sentence pairs with additional monolingual data being used for language modeling, approximately 1.25M English sentences.[4] Unigram, bigram and trigram counts were automatically extracted from the deep syntax English structures and SRILM [14] was used to compute the deep syntax language model. The bilingual deep syntax parsed training data was automatically word-aligned, using our current best performing method for the system: by reconstructing a lemmatized and reordered bilingual corpus from the deep syntactic structures and running Giza++ [15]. Probability distributions for translating factors were computed from relative frequencies of source and target factors of aligned words in the parsed corpus, a selection of which we include in Table 1.[5] In addition, relative frequencies were computed for the factor *case* given the dependency relation between a TL word and its head in the TL structure, a selection of which are shown in Table 2.[6]

All phrasal transfer rules consistent with the word alignment were automatically extracted to compute the deep syntax translation model. Minimum Error Rate Training [16] was carried out on a development set of 500 held-out sentence pairs using Zmert [17] open source tool maximizing for BLEU evaluation metric.

The system was used to translate 1755 German sentences length 5-15 words [2] into English. For parsing, the same German LFG was used as for training and the single-best parse was input to the decoder. Decoding was carried out via a top-down application of deep syntax transfer rules with beam search used to manage the large search space.[7] Hypothesis translations are ranked during decoding using a log-linear combination of feature functions, such as translation model, lexical translation model and deep syntax language model. For each translation, the 100-best target language deep syntactic structures are input to the generator and a single string is generated for each. A standard language model

[3]The parser is non-deterministic, producing all possible parses for each input sentence according to the LFG grammar. We use a disambiguation model to rank parses and only the single-best parse was used for training the MT system.

[4]Again only the single-best parse was used.

[5]Morpho-syntactic factors with $p(v_e|v_f) < 0.01$ are left out.

[6]Morpho-syntactic factors with $p(v_e|d_e) < 0.01$ are left out.

[7]Beam size was set to 100.

**(a)**

$$\underset{\text{die}}{\begin{bmatrix}\ldots\end{bmatrix}}\;\underset{\text{polizei}}{\begin{bmatrix}\text{pers} & 3\\ \text{num} & \text{sg}\\ \text{gend} & \text{fem}\\ \text{case} & \text{nom}\end{bmatrix}}\;\underset{\text{kommen}}{\begin{bmatrix}\text{tense} & \text{past}\\ \text{mood} & \text{ind}\\ \text{passive} & \text{-}\\ \text{c-type} & \text{decl}\end{bmatrix}}\;\rightarrow\;\underset{\text{the}}{\begin{bmatrix}\ldots\end{bmatrix}}\;\underset{\text{police}}{\begin{bmatrix}\text{pers} & 3\\ \text{num} & \text{pl}\\ \text{case} & \text{nom}\end{bmatrix}}\;\underset{\text{come}}{\begin{bmatrix}\text{tense} & \text{past}\\ \text{mood} & \text{ind}\\ \text{passive} & \text{-}\\ \text{c-type} & \text{decl}\\ \text{perf} & \text{-}\\ \text{prog} & +\end{bmatrix}}$$

**(b)**

$$\underset{\text{die}}{\begin{bmatrix}\ldots\end{bmatrix}}\;\underset{\text{polizei}}{\begin{bmatrix}\text{pers} & 3\\ \text{num} & \text{sg}\\ \text{gend} & \text{fem}\\ \text{case} & \text{nom}\end{bmatrix}}\;\underset{\text{kommen}}{\begin{bmatrix}\textbf{tense} & \textbf{pres}\\ \text{mood} & \text{ind}\\ \text{passive} & \text{-}\\ \text{c-type} & \text{decl}\end{bmatrix}}\;\rightarrow\;\underset{\text{the}}{\begin{bmatrix}\ldots\end{bmatrix}}\;\underset{\text{police}}{\begin{bmatrix}\text{pers} & 3\\ \text{num} & \text{pl}\\ \text{case} & \text{nom}\end{bmatrix}}\;\underset{\text{come}}{\begin{bmatrix}\textbf{tense} & \textbf{?}\\ \text{mood} & \text{ind}\\ \text{passive} & \text{-}\\ \text{c-type} & \text{decl}\\ \text{perf} & \text{-}\\ \text{prog} & +\end{bmatrix}}$$

Figure 3: Factor template example correctly handling an idiosyncratic translation: **(a)** factor template for the German-English lemmatized phrase pair: *die polizei kommen*$|||$*the police come*, **(b)** factored SL input phrase *Die Polizei kommt* correctly translates *Polizei* from singular in German into plural in English, mismatching features in the source input are in bold and factors translated separately from the lemma have value '?'

score is computed for each of the 100-best output strings for each translation, as well as a grammaticality feature score using information produced by the generator about the grammaticality of the output string. The decoding features and the post-generation features are combined in a single log-linear model to select the final single-best English translation for each German input sentence.

We include five different methods of translating factored input, (i) plain factored: all factors are translated separately from lemmas using $p(v_e|v_f)$, (ii) factored + *case* special: all factors except *case* are translated separately from lemmas using $p(v_e|v_f)$ and *case* is translated separately from the lemma using $p(v_e|d_e)$, (iii) plain templates: the target side factors in the template of each phrase is used as-is with no factoring, effectively disregarding differences in SL input factors, (iv) templates + mismatching factored: templates are used to translate matching factors and mismatching factors are translated using $p(v_e|v_f)$, (v) templates + mismatching factored + *case* special: templates are used to translate matching factors and all mismatching factors except for *case* are translated using $p(v_e|v_f)$ and *case* is translated using $p(v_e|d_e)$.

### 5.1. Results

Table 3 shows BLEU scores for the MT system using five different methods for translating factors. The results show a low baseline for the plain factored model, in which all factors are translated separately from lemmas, with a BLEU score of 6.23%. Using the dependency relation between a word and its head to translate *case* increases the results slightly to 6.27% BLEU. Taking target side factors directly from the factor templates, with no factors translated separately, results in an improvement, increasing the BLEU score to 8.8%. The two methods that use factor templates to translate all matching SL factors perform best, improving the BLEU score substantially to 16.18% when all factors are translated with $p(v_e|v_f)$, with a small improvement seen when the probability is conditioned on the dependency relation, $p(v_e|d_e)$, for translating *case*, increasing to 16.85% BLEU.

Table 3 also shows precision, recall and f-score results of translated morpho-syntactic factors when compared to those of the parsed reference translations. The results are in line with the BLEU scores of Table 3, with respect to the rank of each method. For the Factored Models with and without using templates, when we condition the probability used to translate the morpho-syntactic factors on the dependency relation as opposed to the SL factor for *case*, we see no increase in f-score, as the f-score for both configurations without templates is 33% and for both configurations with templates its 41%. The improvement from the baseline plain factored model when compared with the mismatching factor template methods is substantial, from an f-score of 33% to 41%, an increase of 8 percentage points.

Table 4 shows a break-down of translation results for individual morpho-syntactic factors when compared to those of parsed reference translations. The best result for translating each morpho-syntactic factor is achieved using factor templates to translate matching factors while only translating

Table 1: $p(v_e|v_f)$ computed from 360K German-English LFG f-structure pairs

| Morpho-syntactic Factor | $v_f$ | $v_e$ | $p(v_e|v_f)$ |
|---|---|---|---|
| PERSON | 1 | 1 | 0.97 |
| | | 3 | 0.02 |
| | | 2 | 0.01 |
| | 2 | 2 | 0.66 |
| | | 3 | 0.30 |
| | | 1 | 0.04 |
| | 3 | 3 | 0.97 |
| | | 2 | 0.02 |
| | | 1 | 0.01 |
| TENSE | past | present | 0.61 |
| | | past | 0.38 |
| | | future | 0.01 |
| | present | present | 0.88 |
| | | past | 0.06 |
| | | future | 0.06 |
| NUMBER | singular | singular | 0.94 |
| | | plural | 0.06 |
| | plural | plural | 0.86 |
| | | singular | 0.14 |
| CASE | nominative | nominative | 0.85 |
| | | oblique | 0.15 |
| | accusative | oblique | 0.89 |
| | | nominative | 0.11 |
| | dative | oblique | 0.88 |
| | | nominative | 0.12 |
| | genitive | oblique | 0.91 |
| | | nominative | 0.09 |
| PASSIVE | - | - | 0.96 |
| | | + | 0.04 |
| | + | + | 0.74 |
| | | - | 0.26 |
| MOOD | indicative | indicative | 0.99 |
| | subjunctive | indicative | 0.91 |
| | | subjunctive | 0.08 |
| | imperative | indicative | 0.58 |
| | | imperative | 0.42 |

Table 2: $p(v_e|v_f)$ computed from 360K German-English LFG f-structure pairs

| Head Dependency Relation, $d_e$ | Case, $v_e$ | $p(v_e|d_e)$ |
|---|---|---|
| MODIFIER | obl | 1.00 |
| OBJECT | obl | 1.00 |
| THETA OBJECT | obl | 1.00 |
| OBLIQUE AGENT | obl | 1.00 |
| OBLIQUE | obl | 1.00 |
| OBLIQUE PARTICLE | obl | 1.00 |
| INTEROGITIVE PRONOUN | obl | 1.00 |
| TOPIC | obl | 1.00 |
| RELATIVISED TOPIC | obl | 1.00 |
| SUBJECT | nom | 0.99 |
| | obl | 0.01 |
| INTEROGITIVE FOCUS | obl | 0.98 |
| | nom | 0.02 |
| RELATIVE PRONOUN | obl | 0.97 |
| | nom | 0.03 |
| FRAGMENT | nom | 0.96 |
| | obl | 0.04 |
| X-COMPLEMENT | nom | 0.81 |
| | obl | 0.19 |

mismatching factors separately from the lemma.

### 5.2. Discussion

Results show that using templates to translate factors can improve machine translation output significantly. Accurately translating all factors separately from lemmas is not achievable due to the very large number of possible combinations of values and the fact that generation in the deep syntactic transfer SMT system is carried out on the sentence level. This results in only 100 translation options being generated per SL input sentence, a very low portion considering the large number of translation options per word. Using factor templates to translate factors that match the source input factors results in much higher BLEU scores and morpho-syntactic precision and recall scores when compared to reference translations.

Examining the probability distributions computed from the word-aligned corpus for translating individual morpho-syntactic factors reveals some interesting insights into how well factors correspond between the German and English words of the corpus. In Table 1, the probability distribution for *person* shows that in the training data, only 66% of nouns in the $2^{nd}$ person in German are translated into English as the $2^{nd}$ person, with 30% being translated as $3^{rd}$ person and 4% being translated as $1^{st}$ person. Another surprising statistic shows up in Table 1 for *tense*, that 61% of verbs in the past tense are translated into the present tense in English, with a smaller amount being translated as *past*, 38%, and a very small number as *future* 1%. However, its worth mentioning

Table 3: Automatic evaluation results on 1755 held-out German-English sentence pairs

| Factor Translation | BLEU | Precision | Recall | F-score |
|---|---|---|---|---|
| plain factored | 6.23 | 34 % | 32 % | 33 % |
| factored + *case* special | 6.27 | 34 % | 32 % | 33 % |
| plain templates | 8.80 | 35 % | 33 % | 34 % |
| templates + mismatching factored | 16.18 | 40 % | 41 % | 41 % |
| templates + mismatching factored + *case* special | 16.85 | 40 % | 41 % | 41 % |

that *tense* at the f-structure level of analysis in LFG is not simply divided notionally into *past*, *present* and *future*. For example, the tense of the German verb *gehen* in *Ich ging* is analyzed as:

- *tense past*, mood indicative,

and the corresponding verb in its English translation *I went* is given a similar analysis for tense:

- *tense past*, progressive -, perfect -, mood indicative,

but the alternate translation *I have gone* is analyzed as follows

- *tense present*, progressive -, *perfect* +, mood indicative.

So, although notionally both English translations encode that the event was in the past, syntactically only the former is in the *past* tense, and this phenomenon probably accounts for much of the divergence in *tense* observed in the probability distribution.

In addition, when we examine the probability distribution for *number* in Table 1 we see that a relatively large proportion of nouns that appear in the plural in German are translated into a singular noun in English, 14%. Its not surprising that the values for *case* between German and English don't correspond well, and even when a German noun is in the nominative, only 85% of the time is it translated into the nominative case in English. When translating the *case* of a noun, the dependency relation between a word and its head is more informative than the source language *case* factor, as can be seen from Table 2.

## 6. Conclusions

We presented a method of handling the combinatorial explosion of Factored Machine Translation Models that arises from translating lemmas separately from their morpho-syntactic factors. Our experimental evaluation shows major improvements over the baseline method when tested on a deep syntactic transfer SMT system. We hope that the work presented here will, in the future, can be applied to Phrase-Based Factored Models and show improvements also.

## 7. Acknowledgements

Many thanks to Philipp Koehn for his insight and feedback.

## 8. References

[1] Koehn, P. and Hoang, H., "Factored Translation Models", Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 868-876, 2007.

[2] Koehn, P., Och, F.J. and Marcu, D., "Statistical Phrase-based Translation", Proceedings of Human Language Technology and North American Chapter of the Association for Computational Linguistics Conference, 48-54, 2003.

[3] Riezler, S., Maxwell, J., "Grammatical Machine Translation" Proceedings of Human Language Technologies and the 44th Annual Meeting of the Assciation for Computational Linguistics, New York, 248-255, 2006.

[4] Bojar, O. and Hajič, J., "Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation", Proceedings of the third Workshop on Statistical Machine Translation, Columbus, Ohio, 2008.

[5] Graham, Y., van Genabith, J., Bryl, A., "F-structure Transfer-Based Statistical Machine Translation", Proceedings of the 14th International Lexical Functional Grammar Conference, CSLI Publications, Cambridge, 2009.

[6] Papineni, K., Roukos, S., Ward, T. and Zhu, W., "A Method for Automatic Evaluation of Machine Translation", Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 311-318, 2002.

[7] Kaplan, R. and Bresnan, J., "Lexical Functional Grammar, a Formal System for Grammatical Represenation", The Mental Representation of Grammatical Relations, 173-281, MIT Press, Cambridge, MA. 1982.

[8] Bresnan, J., "Lexical-Functional Syntax", Blackwell Oxford, 2001.

[9] Dalrymple, M., "Lexical Functional Grammar", Academic Press, San Diego, CA; London, 2001.

[10] Koehn, P., "Europarl: A Parallel Corpus for Statistical Machine Translation", Proceedings of the tenth Machine Translation Summit., 2005.

[11] Kaplan, R.M., King, T.H. and Maxwell, J.T., "Adapting Existing Grammars: the XLE Experience", Proceedings of the 19th International Conference on Computational Linguistics (COLING) 2002, Taipei, Taiwan, 2002.

[12] Kaplan, R.M., Riezler, S., King, T.H., Maxwell J.T. and Vasserman A., "Speed and Accuracy in Shallow and Deep Stochastic Parsing", Proceedings of Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Meeting, Boston, MA, 2004.

[13] Riezler, S., King, T.H., Kaplan, R.M., Crouch, R., Maxwell, J.T. and Johnson, M., "Parsing the Wall Street Journal using Lexical Functional", Grammar and Discriminitive Estimation Techniques . Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL), Philadelphia, 2002.

[14] Stolcke, A., "SRILM - An Extensible Language Modeling Toolkit", Proceedings of the International Conference on Spoken Language Processing, Denver, Colorado, 2002.

[15] Och, F. J., Tillmann, C. and Ney, H., "Improved Alignment Models for Statistical Machine Translation", Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 20-28, College Park, MD, 1999.

[16] Och, F. J., "Minimum Error Rate Training in Statistical Machine Translation", Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 160-167, Sapporo, Japan, 2003.

[17] Zaidan, O., "Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems", The Prague Bulletin of Mathematical Linguistics, No. 91:79-88, 2009.

Table 4: Comparison of a selection of automatically translated factors and reference translation factors

| Morpho-syntactic Factor | Factor Translation Method | Precision | Recall | F-score |
|---|---|---|---|---|
| CASE | factored | 40 % | 39 % | 40 % |
| | factored + *case* special | 41 % | 40 % | 40 % |
| | template | 38 % | 36 % | 37 % |
| | template + mismatching factored | 45 % | 48 % | 46 % |
| | template + mismatching factored + *case* special | 45 % | 47 % | 46 % |
| PERSON | factored | 49 % | 42 % | 45 % |
| | template | 50 % | 43 % | 46 % |
| | template + mismatching factored | 54 % | 54 % | 54 % |
| TENSE | factored | 38 % | 31 % | 34 % |
| | template | 37 % | 30 % | 33 % |
| | template + mismatching factored | 39 % | 33 % | 36 % |
| NUMBER | factored | 48 % | 41 % | 44 % |
| | template | 46 % | 39 % | 42 % |
| | template + mismatching factored | 53 % | 53 % | 53 % |
| PASSIVE | factored | 39 % | 31 % | 34 % |
| | template | 40 % | 32 % | 35 % |
| | template + mismatching factored | 42 % | 36 % | 39 % |
| MOOD | factored | 42 % | 34 % | 38 % |
| | template | 43 % | 34 % | 38 % |
| | template + mismatching factored | 45 % | 38 % | 41 % |