A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation

Rico Sennrich and Barry Haddow
School of Informatics, University of Edinburgh
rico.sennrich@ed.ac.uk, bhaddow@inf.ed.ac.uk

Abstract

When translating between two languages that differ in their degree of morphological synthesis, syntactic structures in one language may be realized as morphological structures in the other, and SMT models need a mechanism to learn such translations. Prior work has used morpheme splitting with flat representations that do not encode the hierarchical structure between morphemes, but this structure is relevant for learning morphosyntactic constraints and selectional preferences. We propose to model syntactic and morphological structure jointly in a dependency translation model, allowing the system to generalize to the level of morphemes. We present a dependency representation of German compounds and particle verbs that results in improvements in translation quality of 1.4–1.8 BLEU in the WMT English–German translation task.

1 Introduction

When translating between two languages that differ in their degree of morphological synthesis, syntactic structures in one language may be realized as morphological structures in the other. Machine Translation models that treat words as atomic units have poor learning capabilities for such translation units, and morphological segmentations are commonly used (Koehn and Knight, 2003). Like words in a sentence, the morphemes of a word have a hierarchical structure that is relevant in translation. For instance, compounds in Germanic languages are head-final, and the head is the segment that determines agreement within the noun phrase, and is relevant for selectional preferences of verbs.

1. sie erheben eine Handlungsgebühr.

In example 1, agreement in case, number and gender is enforced between eine ’a’ and Gebühr ’fee’, and selectional preference between erheben ’charge’ and Gebühr ’fee’. A flat representation, as is common in phrase-based SMT, does not encode these relationships, but a dependency representation does so through dependency links.

In this paper, we investigate a dependency representation of morphologically segmented words for SMT. Our representation encodes syntactic and morphological structure jointly, allowing a single model to learn the translation of both. Specifically, we work with a string-to-tree model with GHKM-style rules (Galley et al., 2006), and a relational dependency language model (Sennrich, 2015). We focus on the representation of German compounds and particle verbs.

German makes heavy use of compounding, and compounds such as Abwasserbehandlungsanlage ‘waste water treatment plant’ are translated into complex noun phrases in other languages, such as French station d’épuration des eaux résiduaires.

German particle verbs are difficult to model because their surface realization differs depending on the finiteness of the verb and the type of clause. Verb particles are separated from the finite verb in...
main clauses, but prefixed to the verb in subordinated clauses, or when the verb is non-finite. The infinitive marker *zu* ‘to’, which is normally a pre-modifying particle, appears as an infix in particle verbs. Table 1 shows an illustrating example.

## 2 A Dependency Representation of Compounds and Particle Verbs

The main focus of research on compound splitting has been on the splitting algorithm (Popovic et al., 2006; Nießen and Ney, 2000; Weller et al., 2014; Macherey et al., 2011). Our focus is not the splitting algorithm, but the representation of compounds. For splitting, we use an approach similar to (Fritzinger and Fraser, 2010), with segmentation candidates identified by a finite-state morphology (Schmid et al., 2004; Sennrich and Kunz, 2014), and statistical evidence from the training corpus to select a split (Koehn and Knight, 2003).

German compounds are head-final, and pre-modifiers can be added recursively. Compounds are structurally ambiguous if there is more than one modifier. Consider the distinction between *(Stadtteil)projekt* (literally: ‘(city part) project’) and *Stadt(teilprojekt)* ‘city sub-project’. We opt for a left-branching representation by default.\(^1\) We also split linking elements, and represent them as a postmodifier of each non-final segment, including the empty string (“\(\epsilon\)”). We use the same representation for noun compounds and adjective compounds.

An example of the original\(^2\) and the proposed compound representation is shown in Figure 1. Importantly, the head of the compound is also the parent of the determiners and attributes in the noun phrase, which makes a bigram dependency language model sufficient to enforce agreement. Since we model morphosyntactic agreement within the main translation step, and not in a separate step as in (Fraser et al., 2012), we deem it useful that inflection is marked at the head of the compound. Consequently, we do not split off inflectional or derivational morphemes.

For German particle verbs, we define a common representation that abstracts away from the various surface realizations (see Table 1). Separated

---

\(^1\)We follow prior work in leaving frequent words or sub-words unsplit, which has a disambiguating effect. With more aggressive splitting, frequency information could be used for the structural disambiguation of internal structure.

\(^2\)The original dependency trees follow the annotation guidelines by Foth (2005).

---

![Figure 1: Original and proposed representation of German compound.](image1)

![Figure 2: Original and proposed representation of German particle verb with infixed *zu*-marker.](image2)
pendency language model (RDLM). Figure 3 (a) shows the constituency representation of the example in Figure 1.

Our model should not only be able to produce new words productively, but also to memorize words it has observed during training. Looking at the compound Handgepäckgebühr in Figure 3 (a), we can see that it does not form a constituent, and cannot be extracted with GHKM extraction heuristics. To address this, we binarize the trees in our training data (Wang et al., 2007).

A complicating factor is that the binarization should not impair the RDLM. During decoding, we map the internal tree structure of each hypothesis back to the unbinarized form, which is then scored by the RDLM. Virtual nodes introduced by the binarization must also be scorable by RDLM if they form the root of a translation hypothesis. A simple right or left binarization would produce virtual nodes without head and without meaningful dependency representation. We ensure that each virtual node dominates the head of the full constituent through a mixed binarization.\(^3\) Specifically, we perform right binarization of the head and all pre-modifiers, then left binarization of all post-modifiers. This head-binarized representation is illustrated in Figure 3 (b).\(^4\)

Head binarization ensures that even hypotheses whose root is a virtual node can be scored by the RDLM. This score is only relevant for pruning, and discarded when the full constituent is scored. Still, these hypotheses require special treatment in the RDLM to mitigate search errors. The virtual node labels (such as OBJA) are unknown symbols to the RDLM, and we simply replace them with the original label (OBJA). The RDLM uses sibling context, and this is normally padded with special start and stop symbols, analogous to BOS/EOS symbols in \(n\)-gram models. These start and stop symbols let the RDLM compute the probability that a node is the first or last child of its ancestor node. However, computing these probabilities for virtual nodes would unfairly bias the search, since the first/last child of a virtual node is not necessarily the first/last child of the full constituent. We adapt the representation of virtual nodes in RDLM to take this into account. We distinguish between virtual nodes based on whether their span is a string prefix, suffix, or infix of the full constituent. For prefixes and infixes, we do not add a stop symbol at the end, and use null symbols, which denote unavailable context, for padding to the right. For suffixes and infixes, we do the same at the start.

4 Post-Processing

For SMT, all German training and development data is converted into the representation described in sections 2–3. To restore the original representation, we start from the tree output of the string-to-tree decoder. Merging compounds is trivial: all segments and linking elements can be identified by the tree structure, and are concatenated.

For verbs that dominate a verb particle, the original order is restored through three rules:

1. non-finite verbs are concatenated with the particle, and zu-markers are infixed.
2. finite verbs that head a subordinated clause (identified by its dependency label) are concatenated with the particle.
3. finite verbs that head a main clause have the

\(^3\)In other words, every node is a fixed well-formed dependency structure (Shen et al., 2010) with our binarization.

\(^4\)Note that our definition of head binarization is different from that of Wang et al. (2007), who left-binarize a node if the head is the first child, and right-binarize otherwise. Our algorithm also covers cases where the head has both pre- and post-modifiers, as erheben and gepack do in Figure 3.
particle moved to the right clause bracket.\textsuperscript{5}

Previous work on particle verb translation into German proposed to predict the position of particles with an $n$-gram language model (Nießen and Ney, 2001). Our rules have the advantage that they are informed by the syntax of the sentence and consider the finiteness of the verb.

Our rules only produce projective trees. Verb particles may also appear in positions that violate projectivity, and we leave it to future research to determine if our limitation to projective trees affects translation quality, and how to produce non-projective trees.

5 SMT experiments

5.1 Data and Models

We train English–German string-to-tree SMT systems on the training data of the shared translation task of the Workshop on Statistical Machine Translation (WMT) 2015. The data set consists of 4.2 million sentence pairs of parallel data, and 160 million sentences of monolingual German data.

We base our systems on that of Williams et al. (2014). It is a string-to-tree GHKM translation system implemented in Moses (Koehn et al., 2007), and using the dependency annotation by ParZu (Sennrich et al., 2013). Additionally, our baseline system contains a dependency language model (RDLM) (Sennrich, 2015), trained on the target-side of the parallel training data.

We report case-sensitive B\textsc{leu} scores on the newstest2014/5 test sets from WMT, averaged over 3 optimization runs of k-batch MIRA (Cherry and Foster, 2012) on a subset of newstest2008-12.\textsuperscript{6}

We split all particle verbs and hyphenated compounds, but other compounds are only split if they are rare (frequency in parallel text < 5).

For comparison with the state-of-the-art, we train a full system on our restructured representation, which incorporates all models and settings of our WMT 2015 submission system (Williams et al., 2014).\textsuperscript{7} Note that our WMT 2015 submission uses the dependency representation of compounds and tree binarization introduced in this paper; we achieve additional gains over the submission system through particle verb restructuring.

5.2 SMT Results

Table 2 shows translation quality (B\textsc{leu}) with different representations of German compounds and particle verbs. Head binarization not only yields improvements over the baseline, but also allows for larger gains from morphological segmentation. We attribute this to the fact that full compounds, and prefixed particle verbs, are not always a constituent in the segmented representation, and that binarization compensates this theoretical drawback.

With head binarization, we find substantial improvements from compound splitting of 0.7–1.1 BLEU. On newstest2014, the improvement is almost twice of that reported in related work (Williams et al., 2014), which also uses a hierarchical representation of compounds, albeit one that does not allow for dependency modelling. Examples of correct, unseen compounds generated include Staubsauger\textsubscript{roboter} 'vacuum cleaner robot', Gravitations\textsubscript{wellen} 'gravitational waves', and NPD-l\textsubscript{verbot}l\textsubscript{verfahren} 'NPD banning process'.\textsuperscript{8}

8We use \texttt{mteval-v13a.pl} for comparability to official WMT results; all significance values reported are obtained with MultiEval (Clark et al., 2011).

7In contrast to our other systems in this paper, RDLM is trained on all monolingual data for the full system, and two models are added: a 5-gram Neural Network language model

<table>
<thead>
<tr>
<th>system</th>
<th>newstest2014</th>
<th>newstest2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>20.7</td>
<td>22.0</td>
</tr>
<tr>
<td>+split compounds</td>
<td>21.3</td>
<td>22.4</td>
</tr>
<tr>
<td>+particle verbs</td>
<td>21.4</td>
<td>22.8</td>
</tr>
<tr>
<td>head binarization</td>
<td>20.9</td>
<td>22.7</td>
</tr>
<tr>
<td>+split compounds</td>
<td>22.0</td>
<td>23.4</td>
</tr>
<tr>
<td>+particle verbs</td>
<td>22.1</td>
<td>23.8</td>
</tr>
<tr>
<td>full system</td>
<td>22.6</td>
<td>24.4</td>
</tr>
</tbody>
</table>

Table 2: English–German translation results (B\textsc{leu}). Average of three optimization runs.

<table>
<thead>
<tr>
<th>system</th>
<th>compound</th>
<th>particle verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>2841</td>
<td>533</td>
</tr>
<tr>
<td>baseline</td>
<td>845</td>
<td>96</td>
</tr>
<tr>
<td>+head binarization</td>
<td>798</td>
<td>157</td>
</tr>
<tr>
<td>+split compounds</td>
<td>1850</td>
<td>160</td>
</tr>
<tr>
<td>+particle verbs</td>
<td>1992</td>
<td>333</td>
</tr>
</tbody>
</table>

Table 3: Number of compounds [that would be split by compound splitter] and particle verbs (separated, prefixed and with zu-infix) in newstest2014/5. Average of three optimization runs.
Particle verb restructuring yields additional gains of 0.1–0.4 BLEU. One reason for the smaller effect of particle verb restructuring is that the difficult cases – separated particle verbs and those with infixation – are rarer than compounds, with 2841 rare compounds [that would be split by our compound splitter] in the reference texts, in contrast to 553 separated particle verbs, and 176 particle verbs with infixation, as Table 3 illustrates. If we only evaluate the sentences containing a particle verb with zu-infix in the reference, 165 in total for newstest2014/5, we observe an improvement of 0.8 BLEU on this subset (22.1 → 22.9), significant with $p < 0.05$.

The positive effect of restructuring is also apparent in frequency statistics. Table 3 shows that the baseline system severely undergenerates compounds and separated/infixed particle verbs. Binarization, compound splitting, and particle verb restructuring all contribute to bringing the distribution of compounds and particle verbs closer to the reference.

In total, the restructured representation yields improvements of 1.4–1.8 BLEU over our baseline. The full system is competitive with official submissions to the WMT 2015 shared translation tasks. It outperforms our submission (Williams et al., 2015) by 0.4 BLEU, and outperforms other phrase-based and syntax-based submissions by 0.8 BLEU or more. The best reported result according to BLEU is an ensemble of Neural MT systems (Jean et al., 2015), which achieves 24.9 BLEU. In the human evaluation, both our submission and the Neural MT system were ranked 1–2 (out of 16), with no significant difference between them.

### 5.3 Synthetic LM Experiment

We perform a synthetic experiment to test our claim that a dependency representation allows for the modelling of agreement between morphemes. For 200 rare compounds [that would be split by our compound splitter] in the newstest2014/5 references, we artificially introduce agreement errors by changing the gender of the determiner. For instance, we create the erroneous sentence *sie erheben ein Handgepäckgebühr* as a complement to Example 1. We measure the ability of language models to prefer (give a higher probability to) the original reference sentence over the erroneous one. In the original representation, both a Kneser-Ney 5-gram LM and RDLM perform poorly due to data sparseness, with 70% and 57.5% accuracy, respectively. In the split representation, the RDLM reliably prefers the correct agreement (96.5% accuracy), whilst the performance of the 5-gram model even deteriorates (to 60% accuracy). This is because the gender of the first segment(s) is irrelevant, or even misleading, for agreement. For instance, *Handgepäck* is neuter, which could lead a morpheme-level n-gram model to prefer the determiner *ein*, but *Handgepäckgebühr* is feminine and requires *eine*.

### 6 Conclusion

Our main contribution is that we exploit the hierarchical structure of morphemes to model them jointly with syntax in a dependency-based string-to-tree SMT model. We describe the dependency annotation of two morphologically complex word classes in German, compounds and particle verbs, and show that our tree representation yields improvements in translation quality of 1.4–1.8 BLEU in the WMT English–German translation task.\(^9\)

The principle of jointly representing syntactic and morphological structure in dependency trees can be applied to other language pairs, and we expect this to be helpful for languages with a high degree of morphological synthesis. However, the annotation needs to be adapted to the respective languages. For example, French compounds such as *arc-en-ciel* ‘rainbow’ are head-initial, in contrast to head-final Germanic compounds.

### Acknowledgments

This project received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), 644402 (HimL), 644333 (TraMOOC), and from the Swiss National Science Foundation under grant P2ZHP1_148717.

### References


\(^9\)We released source code and configuration files at https://github.com/rsennrich/wmt2014-scripts.


