SMT of German Patents at WIPO: Decompounding and Verb Structure Pre-reordering

Marcin Junczys-Dowmunt
Adam Mickiewicz University
Information Systems Laboratory
ul. Umultowska 87
61-614 Poznań, Poland
junczys@amu.edu.pl

Bruno Pouliquen
World Intellectual Property Organization
Global Database Service
34, chemin des Colombettes
CH-1211 Geneva
bruno.pouliquen@wipo.int

Abstract

We describe fragments of the SMT pipeline at WIPO for German as a source language. Two subsystems are discussed in detail: word decompounding and verb structure pre-reordering. Apart from automatic evaluation results for both subsystems, for the pre-reordering mechanism manual evaluation results are reported.

1 Introduction

German is one of the 10 official publication languages in which a Patent application can be filed at WIPO [1]. Among the European languages, German proves to be the most challenging one for WIPO's in-house SMT system.

In contrast to French, English, or Spanish, extensive preprocessing has to be applied when German is the source language. In this paper we will illustrate fragments of the Patent SMT pipeline deployed at WIPO that deal with these problems (Pouliquen and Mazenc, 2011). Decompounding has been an established part of the WIPO pipeline, verb structure pre-reordering is a recent addition.

2 German Compound Words

German has the particularity to join individual words into compound words. This is a challenge for SMT as it generates OOV words and data sparseness. Especially patents "suffer" from compound words, e.g. a recent German patent was titled "gasballongetragener flugroboter"[2], both words previously unseen. To solve this problem we apply a "decompounding" process ("gas∼ ballon∼ getragener flug∼ roboter") before training and then proceed with the standard SMT training process.

2.1 Related Work

Koehn and Knight (2003) use parallel texts to train a compound splitter: after aligning the segments, they search for possible splits where each part has a translation as one word in the target segment. POS-information is used as a filter. Popović et al. (2006) experiment with two compound splitting methods (German-English): linguistic and corpus-based and reach similar results for both methods. Junczys-Dowmunt (2008) proposes high-accuracy methods for compound splitting.

At WIPO, decompounding is also used in the in-house developed tools for patent search, CLIR and PATENTSCOPE (Pouliquen and Mazenc, 2011). As our goal is two-fold (SMT and IR), we have to increase precision and recall of our decompounder. Leveling et al. (2011) mentions "Patents have a specific writing style and vocabulary", so we adopt a bottom-up approach learning compound words from the available parallel data. As we plan to use the tool for other languages in the future, no POS information is used.

2.2 Method

We train an SMT system on our parallel English-German data (1.8M segments, 570M English words) and use phrase tables entries as input for the following "compound word guessing" process:

[1] The 10 publication languages under the Patent Cooperation Treaty are Arabic, Chinese, English, French, German, Japanese, Korean, Portuguese, Russian and Spanish.

[2] "gas balloon carried flight robot". We will refer to all German compound words using lowercase letters.

1. Create a German-English dictionary of 1-1 entries (eg. “roboter” → “robot” “gas” → “gas”, “flug” → “flight”) at a probability threshold of 0.01.
2. Create a dictionary of 1-2 entries (eg. “flugroboter” → “flight robot”)
3. Check that segments of two-part decompositions have translations in the 1-1 dictionary (ie. “flug∼ roboter” “flug” → “flight” and “roboter” → “robot”), allowing for “filler” letters like “s” or “er” (e.g “publikations∼ programm”)
4. Create a dictionary of 1-3 entries of compound words used as prefixes (“flugroboterprogramm” → “flight robot program”)
5. Repeat from 3) until no more compound words can be learned.

German compounds words that are commonly translated as single English words are blacklisted (i.e. the “neu∼ ordnung” can be decomposed as “new order”, but “reorganization” is preferred). This blacklist can contain false negatives if English words are compound words themselves, e.g. “roll∼ stühle” → “wheelchair”. Therefore we also check against a German→French list (“roll∼ stühle” → “fauteuil roulant”). To increase the list of compounds we repeat the process for our German-French and join both lists. With time, many compounds are added manually to that list. This results in a list of 644,275 compound words.

The decompounding algorithm is straightforward: we decompose the given word in seen compound words or seen compound segments. A last filter is applied: we check that the average segment length is at least 3.5 characters (avoiding decompositions like “co∼ de∼ bit” for “codebit”).

So far, the longest compound word found in our corpus is “verteil∼ vorrichtung∼ luft∼ strömungs∼ wärme∼ regulierungs∼ kreislauf∼ element∼ kennzeichnungs∼ system”.

2.3 Evaluation

Decompounding is evaluated for English and German in both directions on a small subcorpus of 1 million segments (42 million English words). Table 1 summarizes these results. We observe an improvement of 3 to 4 points BLEU in both cases.

Direction	W/o decomp. BLEU	With decomp. BLEU
en→de	35.18	38.01
de→en	44.86	48.85

Table 1: Automatic evaluation for decompounding

```
V.FIN * V.(PP|INF) → 1 3 2
V.FIN * PTKVZ → 3 1 2
ˆ KON * PTKZU V.INF → 1 3 4 2
```

Figure 1: Reordering Example Rules

3 Verb Structure Pre-Reordering

German clause structures pose another difficult problem. Often the meaningful part of a German verbal complex appears at the end of the sentence. Patents seem to favour long sentences. Thus, the meaningful verb part may appear at the end of a long sentence, many words away from the subject. Phrase-based SMT is not capable of capturing such long-distance relationships and often fails to translate the verb entirely.

3.1 Related Work

Many approaches for clause restructuring exist, we only refer to a few. For German, Collins et. al (2005) describe a syntactic parsing approach with manually written reordering rules for the parsed trees. Reordering rules inferred automatically from parse trees and word alignments, have been proposed for Chinese (Li et al., 2007).

Syntactic parsing is resource-hungry and time-intensive and cannot be part of our pipeline. Less demanding approaches rely on part-of-speech taggers, see Popović and Ney (2006) for manually written rules or Niehues and Kolss (2009) for automatically induced reordering pattern.

3.2 Our Method

Our approach is a shallow one with manually written rules that rely on POS tags. These rules are combined with selection algorithms that are based on alignment data or if alignment data is unavailable on a maximum entropy classifier. Both, part-of-speech tagger and the maximum entropy classifier, are part of the open-source package Apache OpenNLP[3]. Figure 1 contains a few example rules. The first part consists of regular-expression-like pattern that has to be matched by the POS-tagged sentence. The second part illustrates the reordering operation. Numbers correspond to positions of matched tokens in the pattern.

[3] http://opennlp.apache.org

3.2.1 Alignment-based Reordering Selection

Alignment-based rule selection can only be applied during translation model training. The training procedure is interrupted after word-alignment symmetrization and before phrase table extraction. The source training corpus is reordered and the corresponding alignment is modified to match the newly reordered German sentences.

Algorithm 1 is applied to a source sentence s and the corresponding alignment A. The function $\mathrm{matchingRules}$ returns a set of candidate reordering rules applicable to s. Each subset of rules is applied to the input sentence and the input alignment ($\mathcal{P}(M)$ is the powerset of the set of all rules). If the reordered alignment scores better according to $\mathrm{linedist}$ than the previous best reordering, the new best reordered sentence, alignment and rule set are preserved. At the end, the overall best candidates are returned. Candidate reorderings are scored based on the distance of the reordered alignment from an idealized line (linear least squares):

$$\begin{aligned} a &= \min\{i|(i,j) \in A\} \\ b &= \max\{i|(i,j) \in A\} \\ c &= \min\{j|(i,j) \in A\} \\ d &= \max\{j|(i,j) \in A\} \end{aligned}$$

$$\mathrm{linedist}(A) = \sum_{(i,j)\in A} \left(j - \frac{d-c}{b-a}(i-a) + c \right)^2$$

The smaller the distance the more similar is the word order of source and target sentence. Rules in a rule set may be mutually exclusive or overlapping. In that case the rules with the largest matching span take precedence over other rules.

3.2.2 Classifier-based Reordering Selection

During deployment, alignment data is unavailable for unseen sentences and we replace the alignment information with a probabilistic classifier.

The binary maximum entropy classifier used decides whether a rule should be applied (“YES”) or not (“NO”). Samples are collected during the translation model training step described above. Figure 2 shows three example samples, table 2 contains applied the feature types. Applied rules are assigned a “YES” all other rules “NO”.

Algorithm 2 illustrates the application of the classifier. Matching rules for a German source sentence are identified and features for each rule are generated. If the probability of rule application is higher than the probability of the opposite case the rule is kept and applied to the sentence.

```
Input:
$s$ – source sentence (POS-tagged);
$A$ – word alignment; $R$ – reordering rules;
Output:
Best reordered sentence, alignment, applied rules.
begin
    $\hat{s} \leftarrow s$; $\hat{A} \leftarrow A$; $\hat{M} \leftarrow \emptyset$
    $M \leftarrow \mathrm{matchingRules}(s, R)$
    foreach $M' \in \mathcal{P}(M)$ do
        $(s', A') \leftarrow \mathrm{reorder}(s, A, M')$
        if $\mathrm{linedist}(A') < \mathrm{linedist}(\hat{A})$ then
            $\hat{s} \leftarrow s'$; $\hat{A} \leftarrow A'$; $\hat{M} \leftarrow M'$
        end
    end
    return $(\hat{s}, \hat{A}, \hat{M})$
end
```

Algorithm 1: Reordering by alignment

3.3 Automatic and Manual Evaluation

We favour a high precision tool that should not modify a sentence if it might decrease translation quality. The percentage of reordered sentences varies is 5% to 15%. Improvements in BLEU on the test set (1000 sentences) are moderate, but persist when weights are exchanged between optimization runs with and without pre-reordering to exclude optimizer instability. BLEU results for our systems are reported in Tab. 3, “All” is the full test set, “Diff.” reordered sentences (79/1000).

We perform a quick manual evaluation on the 79 changed sentences (Tab. 4). All sentences are evaluated in form of a tournament. Given the source sentence and two outputs, the evaluator declares a win or a draw. System outputs are shuffled,

Feature	Description
`name`	Current rule name
`spanN`	Matched symbol spans
`prevtag`	POS-tag preeceding match
`nexttag`	POS-tag following match
`symN`	Matched rule symbols
`*tagN`	POS-tags spanned by *
`other`	Other possible rules

Table 2: Feature types used

```
NO name=^_⋆_VVIZU_::_2_1 span0=(0,3) span1=(4,4) nexttag=ADJA
  other=PRELS_⋆_V.⋆?_::_1_3_2 sym0=^ ⋆tag0=ART ⋆tag0=NN sym0=⋆ sym1=VVIZU
YES name=PRELS_⋆_V.⋆?_::_1_3_2 span0=(12,12) span1=(13,17) span2=(18,18)
  nexttag=$, prevtag=$, other=^_⋆_VVIZU_::_2_1 sym0=PRELS ⋆tag0=ART ⋆tag0=ADJA
  sym1=⋆ sym2=V.⋆?
NO name=PRELS_⋆_V.⋆?_::_1_3_2 span0=(27,27) span1=(28,29) span2=(30,30)
  nexttag=APPR prevtag=$, sym0=PRELS ⋆tag0=NN ⋆tag1=APPR sym1=⋆ sym2=V.⋆?
```

Figure 2: Samples used for classifier training, first element is class.

```
Input:
$s$ – source sentence (POS-tagged);
$C$ – ME classifier; $R$ – reordering rules;
Output:
Best-scored reordered sentence, applied rules.
begin
    $\hat{M} \leftarrow \emptyset$; $M \leftarrow \text{matchingRules}(s, R)$
    foreach $m \in M$ do
        $\omega \leftarrow \text{features}(s, m, M)$
        if $P_C(\text{YES}|\omega) > P_C(\text{NO}|\omega)$ then
            $\hat{M} \leftarrow \hat{M} \cup \{m\}$
        end
    end
    $s' \leftarrow \text{reorder}(s, \hat{M})$
    return $(\hat{s}, \hat{M})$
end
```

Algorithm 2: Reordering by classifier

the evaluator is unaware which system produced which output. 26 sentences were translated better than their original counterpart, 10 worse and 43 equally good or bad. Among those equally rated 43 sentences, 13 translation were identical.

4 Conclusions

We presented parts of the WIPO patent machine translation pipeline that deal with translation from German. We show that good-practice methods applied in research (e.g. at WMT) can be successfully transferred into user settings (The described method is now in production and publically accessible at: `http://patentscope.wipo.int/translate/`). Decompounding for German achieves good results even with frequent oversplitting. Verb structure reordering is currently very conservative and has only a small but nevertheless beneficial effect on translation from German into other languages.

System	All	Diff.
Baseline	44.91	39.21
Pre-reordered	45.18	41.15

Table 3: BLEU for all and changed sentences

Total	Better	Worse	Equal
79	26 (33%)	10 (13%)	43 (54%)

Table 4: Manual evaluation of pre-reordered sentences compared to original sentences

References

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proc. of ACL*, pages 531–540.

Junczys-Dowmunt, Marcin. 2008. Influence of Accurate Compound Noun Splitting on Bilingual Vocabulary Extraction. In *Proc. of Konvens)*, pages 91–105.

Koehn, Philipp and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proc. of EACL*, pages 187–193.

Leveling, Johannes, Walid Magdy, and Gareth J. F. Jones. 2011. An Investigation of Decompounding for Cross-Language Patent Search. In *Proc. of ACM SIGIR*, pages 1169–1170.

Li, Chi-Ho, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation. In *ACL*, pages 720–727.

Niehues, Jan, Muntsin Kolss, and Universitt Karlsruhe. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Proc. of WMT*, pages 206–214.

Popović, Maja and Hermann Ney. 2006. POS-based Word Reorderings for Statistical Machine Translation. In *Proc. of LREC*, pages 1278–1283.

Popović, Maja, Daniel Stein, and Hermann Ney. 2006. Statistical Machine Translation of German Compound Words. In *Proc. of FinTAL*, pages 616–624.

Pouliquen, Bruno and Christophe Mazenc. 2011. COPPA, CLIR and TAPTA: three tools to assist in overcoming the patent barrier at WIPO. In *Proc of MT-Summit XIII*, pages 24–30, Xiamen, China.