How to Produce Unseen Teddy Bears: Improved Morphological Processing of Compounds in SMT

Fabienne Cap, Alexander Fraser
CIS, University of Munich
{cap|fraser}@cis.uni-muenchen.de

Marion Weller
IMS, University of Stuttgart
wellermn@ims.uni-stuttgart.de

Aoife Cahill
Educational Testing Service
acahill@ets.org

Abstract

Compounding in morphologically rich languages is a highly productive process which often causes SMT approaches to fail because of unseen words. We present an approach for translation into a compounding language that splits compounds into simple words for training and, due to an underspecified representation, allows for free merging of simple words into compounds after translation. In contrast to previous approaches, we use features projected from the source language to predict compound mergings. We integrate our approach into end-to-end SMT and show that many compounds matching the reference translation are produced which did not appear in the training data. Additional manual evaluations support the usefulness of generalizing compound formation in SMT.

1 Introduction

Productive processes like compounding or inflection are problematic for traditional phrase-based statistical machine translation (SMT) approaches, because words can only be translated as they have occurred in the parallel training data. As parallel training data is limited, it is desirable to extract as much information from it as possible. We present an approach for compound processing in SMT, translating from English to German, that splits compounds prior to training (in order to access the individual words which together form the compound) and recombines them after translation. While compound splitting is a well-studied task, compound merging has not received as much attention in the past. We start from Stymne and Cancedda (2011), who used sequence models to predict compound merging and Fraser et al. (2012) who, in addition, generalise over German inflection. Our new contributions are: (i) We project features from the source language to support compound merging predictions. As the source language input is fluent, these features are more reliable than features derived from target language SMT output. (ii) We reduce compound parts to an underspecified representation which allows for maximal generalisation. (iii) We present a detailed manual evaluation methodology which shows that we obtain improved compound translations.

We evaluated compound processing both on held-out split data and in end-to-end SMT. We show that using source language features increases the accuracy of compound generation. Moreover, we find more correct compounds than the baselines, and a considerable number of these compounds are unseen in the training data. This is largely due to the underspecified representation we are using. Finally, we show that our approach improves upon the previous work.

We discuss compound processing in SMT in Section 2, and summarise related work in Section 3. In Section 4 we present our method for splitting compounds and reducing the component words to an underspecified representation. The merging to obtain German compounds is the subject of Section 5. We evaluate the accuracy of compound prediction on held-out data in Section 6 and in end-to-end SMT experiments in Section 7. We conclude in Section 8.

2 Dealing with Compounds in SMT

In German, two (or more) single words (usually nouns or adjectives) are combined to form a compound which is considered a semantic unit. The rightmost part is referred to as the *head* while all other parts are called *modifiers*. EXAMPLE (1) lists different ways of joining simple words into compounds: mostly, no modification is required (A) or a filler letter is introduced (B). More rarely, a letter is deleted (C), or transformed (D).

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 579–587,
Gothenburg, Sweden, April 26-30 2014.

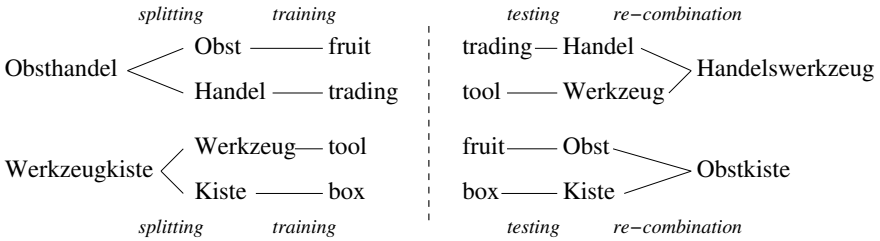

Figure 1: Compound processing in SMT allows the synthesis of compounds unseen in the training data.

EXAMPLE (1)

(A)	*Haus+Boot = Hausboot* ("house boat")
(B)	*Ort+**s**+Zeit = Ort**s**zeit* ("local time")
(C)	*Kirch**e**-**e**+Turm = Kirchturm* ("church tower")
(D)	*Kriteri**um**+Liste = Kriteri**en**liste* ("criteria list")

German compounds are highly productive,[1] and traditional SMT approaches often fail in the face of such productivity. Therefore, special processing of compounds is required for translation into German, as many compounds will not (e.g. *Hausboot*, "house boat") or only rarely have been seen in the training data.[2] In contrast, most compounds consist of two (or more) simple words that occur more frequently in the data than the compound as a whole (e.g. *Haus* (7,975) and *Boot* (162)) and often, these compound parts can be translated 1-to-1 into simple English words. Figure 1 illustrates the basic idea of compound processing in SMT: imagine, *"Werkzeug"* ("tool") occurred only as a modifier of e.g. *"Kiste"* ("box") in the training data, but the test set contains "tool" as a simple word or as the head of a compound. Splitting compounds prior to translation model training enables better access to the component translations and allows for a high degree of generalisation. At testing time, the English text is translated into the split German representation, and only afterwards, some sequences of simple words are (re-)combined into (possibly unseen) compounds where appropriate. This merging of compounds is much more challenging than the splitting, as it has to be applied to disfluent MT output: i.e., compound parts may not occur in the correct word order and even if they do, not all sequences of German words that *could* form a compound *should* be merged.

3 Related Work

Compound processing for translation into a compounding language includes both compound splitting and merging, we thus report on previous approaches for both of these tasks.

In the past, there have been numerous attempts to split compounds, all improving translation quality when translating from a compounding to a non-compounding language. Several compound splitting approaches make use of substring corpus frequencies in order to find the optimal split points of a compound (e.g. Koehn and Knight (2003), who allowed only "(e)s" as filler letters). Stymne et al. (2008) use Koehn and Knight's technique, include a larger list of possible modifier transformations and apply POS restrictions on the substrings, while Fritzinger and Fraser (2010) use a morphological analyser to find only linguistically motivated substrings. In contrast, Dyer (2010) presents a lattice-based approach to encode different segmentations of words (instead of finding the one-best split). More recently, Macherey et al. (2011) presented a language-independent unsupervised approach in which filler letters and a list of words not to be split (e.g., named entities) are learned using phrase tables and Levenshtein distance.

In contrast to splitting, the merging of compounds has received much less attention in the past. An early approach by Popović et al. (2006) recombines compounds using a list of compounds and their parts. It thus never creates invalid German compounds, but on the other hand it is limited to the coverage of the list. Moreover, in some contexts a merging in the list may still be wrong, cf. EXAMPLE (3) in Section 5 below. The approach of Stymne (2009) makes use of a factored model, with a special POS-markup for compound modifiers, derived from the POS of the whole compound. This markup enables sound mergings of compound parts after translation if the POS of the candidate modifier (X-Part) matches the POS of the candidate compound head (X): *Inflations|N-Part + Rate|N = Inflationsrate|N* ("inflation rate"). In Stymne and Cancedda (2011) the factored ap-

[1] Most newly appearing words in German are compounds.

[2] ~30% of the word types and ~77% of the compound types we identified in our training data occurred ≤ 3 times.

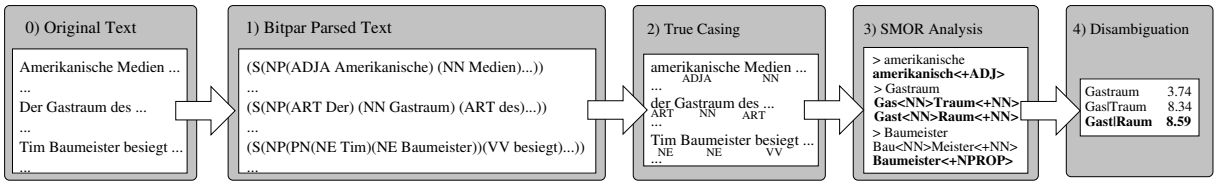

Figure 2: Compound splitting pipeline **1)** The original text is parsed with BITPAR to get unambiguous POS tags, **2)** The original text is then true-cased using the most frequent casing for each word and BITPAR tags are added, **3)** All words are analysed with SMOR, analyses are filtered using BITPAR tags (only **bold**-faced analyses are kept), **4)** If several splitting options remain, the geometric mean of the word (part) frequencies is used to disambiguate them.

proach was extended to make use of a CRF sequence labeller (Lafferty et al., 2001) in order to find reasonable merging points. Besides the words and their POS, many different target language frequency features were defined to train the CRF. This approach can even produce new compounds unseen in the training data, provided that the modifiers occurred in modifier position of a compound and heads occurred as heads or even as simple words with the same inflectional endings. However, as former compound modifiers were left with their filler letters (cf. *"Inflations"*), they can not be generalised to compound heads or simple words, nor can inflectional variants of compound heads or simple words be created (e.g. if *"Rate"* had only been observed in nominative form in the training data, the genitive *"Raten"* could not be produced). The underspecified representation we are using allows for maximal generalisation over word parts independent of their position of occurrence or inflectional realisations. Moreover, their experiments were limited to predicting compounds on held-out data; no results were reported for using their approach in translation. In Fraser et al. (2012) we re-implemented the approach of Stymne and Cancedda (2011), combined it with inflection prediction and applied it to a translation task. However, compound merging was restricted to a list of compounds and parts. Our present work facilitates more independent combination. Toutanova et al. (2008) and Weller et al. (2013) used source language features for target language inflection, but to our knowledge, none of these works applied source language features for compound merging.

4 Step 1: Underspecified Representation

In order to enhance translation model accuracy, it is reasonable to have similar degrees of morphological richness between source and target language. We thus reduce the German target language training data to an underspecified representation: we split compounds, and lemmatise all words (except verbs). All occurrences of simple words, former compound modifiers or heads have the same representation and can thus be freely merged into "old" and "new" compounds after translation, cf. Figure 1 above. So that we can later predict the merging of simple words into compounds and the inflection of the words, we store all of the morphological information stripped from the underspecified representation.

Note that erroneous over-splitting might make the correct merging of compounds difficult[3] (or even impossible), due to the number of correct decisions required. For example, it requires only 1 correct prediction to recombine *"Niederschlag|Menge"* into *"Niederschlagsmenge"* ("amount of precipitation") but 3 for the wrong split into *"nie|der|Schlag|Menge"* ("never|the|hit|amount"). We use the compound splitter of Fritzinger and Fraser (2010), who have shown that using a rule-based morphological analyser (SMOR, Schmid et al. (2004)) drastically reduced the number of erroneous splits when compared to the frequency-based approach of Koehn and Knight (2003). However, we adapted it to work on tokens: some words can, depending on their context, either be interpreted as named entities or common nouns, e.g., *"Dinkelacker"* (a German beer brand or "spelt|field").[4] We parsed the training data and use the parser's decisions to identify proper names, see *"Baumeister"* in Figure 2.

After splitting, we use SMOR to reduce words to lemmas, keeping morphological features like *gender* or *number*, and stripping features like *case*, as illustrated for *"Ölexporteure"* ("oil exporters"):

[3] In contrast, they may not hurt translation quality in the other direction, where phrase-based SMT is likely to learn the split words as a phrase and thus recover from that error.

[4] Note that Macherey et al. (2011) blocked splitting of words which can be used as named entities, independent of context, which is less general than our solution.

No.	Feature Description	Example	Experiment		
			SC	T	TR
1SC	surface form of the word	string: Arbeit<+NN><Fem><Sg>	X	X	
2SC	main part of speech of the word (from the parser)	string: +NN	X	X	
3SC	word occurs in a bigram with the next word	frequency: 0	X	X	
4SC	word combined to a compound with the next word	frequency: 10,000	X	X	X
5SC	word occurs in modifier position of a compound	frequency: 100,000	X	X	
6SC	word occurs in a head position of a compound	frequency: 10,000	X	X	
7SC	word occurs in modifier position vs. simplex	string: P>W (P= 5SC, W= 100,000)	X		
8SC	word occurs in head position vs. simplex	string: S<W (S= 6SC, W= 100,000)	X		
7SC+	word occurs in modifier position vs. simplex	ratio: 10 (10**ceil(log10(5SC/W)))		X	X
8SC+	word occurs in head position vs. simplex	ratio: 1 (10**ceil(log10(6SC/W)))		X	X
9N	different head types the word can combine with	number: 10,000		X	X

Table 1: Target language CRF features for compound merging. **SC** = features taken from Stymne and Cancedda (2011), **SC+** = improved versions, **N** = new feature. Experiments: **SC** = re-implementation of Stymne and Cancedda (2011), **T**= use full **T**arget feature set, **TR** = use **T**arget features, but only a **R**educed set.

EXAMPLE (2)

compound

Öl<NN>Exporteur<+NN><Masc><Nom><Pl>

Öl<+NN>**<Neut><Sg>** Exporteur<+NN>**<Masc><Pl>**

modifier *head*

While the former compound head (*"Exporteure"*) automatically inherits all morphological features of the compound as a whole, the features of the modifier need to be derived from SMOR in an additional step. We need to ensure that the representation of the modifier is identical to the same word when it occurs independently in order to obtain full generalisation over compound parts.

5 Step 2: Compound Merging

After translation from English into the underspecified German representation, post-processing is required to transform the output back into fluent, morphologically fully specified German. First, compounds need to be merged where appropriate, e.g., *"Hausboote"* ("house boats"):

Haus<+NN><Neut><Sg> + *Boot<+NN><Neut><Pl>*
→ *Haus<NN>Boot<+NN><Neut><Pl>* (merged)

and second, all words need to be inflected:

Haus<NN>Boot<+NN><Neut><***Acc***>*<Pl>*
→ *Hausboot**en** (inflected)*

5.1 Target Language Features

To decide which words should be combined, we follow Stymne and Cancedda (2011) who used CRFs for this task. The features we derived from the target language to train CRF models are listed in Table 1. We adapted features No. 1-8 from Stymne and Cancedda (2011). Then, we modified two features (7+8) and created a new feature indicating the productivity of a modifier (9N).

5.2 Projecting Source Language Features

We also use new features derived from the English source language input, which is coherent and fluent. This makes features derived from it more reliable than the target language features derived from disfluent SMT output. Moreover, source language features might support or block merging decisions in unclear cases, i.e., where target language frequencies are not helpful, either because they are very low or they have roughly equal frequency distributions when occurring in a compound (as modifier or head) vs. as a simple word.

In Table 2, we list three types of features:

1. Syntactic features: different English noun phrase patterns that are aligned to German compound candidate words (cf. 10E-13E)
2. The POS tag of the English word (cf. 14E)
3. Alignment features, derived from word alignments (cf. 15E-18E)

The examples given in Table 2 (10E-13E) show that English compounds often have 1-to-1 correspondences to the parts of a German compound. Knowing that two consecutive German simple words are aligned to two English words of the same noun phrase is a strong indicator that the German words should be merged:

EXAMPLE (3)

should be merged:
ein erhöhtes **verkehrs aufkommen** sorgt für chaos "an increased **traffic volume** causes chaos" (S...(**NP**(DT An)(VN increased)(**NN traffic**)(**NN volume**))..)))
should **not** be merged:
für die finanzierung des **verkehrs aufkommen** "**pay** for the financing of **transport**" (VP(**V pay**)(PP(IN for)(NP(NP(DT the)(NN financing)) (PP(IN of)(NP(**NN transport**)..))

In the compound reading of *"verkehr + aufkommen"*, the English parse structure indicates that the words aligned to *"verkehr"* ("traffic") and

No.	Feature Description	Type
10E	word and next word are aligned from a noun phrase in the English source sentence: (NP(NN traffic)(NN accident)) → *Verkehr* ("traffic") + *Unfall* ("accident")	true/false
11E	word and next word are aligned from a gerund construction in the English source sentence: (NP(VBG developing)(NNS nations)) → *Entwicklung* ("development") + *Länder* ("countries")	true/false
12E	word and next word are aligned from a genitive construction in the English source sentence: (NP(NP(DT the)(NN end))(PP(IN of)(NP(DT the)(NN year)) → *Jahr* ("year") + *Ende*("end")	true/false
13E	word and next word are aligned from an adjective noun construction in the English source sentence: (NP (ADJ protective)(NNS measures)) → *Schutz* ("protection") + *Maßnahmen* ("measures")	true/false
14E	print the POS of the corresponding aligned English word	string
15E	word and next word are aligned 1-to-1 from the same word in the English source sentence, e.g., $beef \nearrow \frac{Rind_{(\text{“}cow\text{”})}}{Fleisch_{(\text{“}meat\text{”})}} \searrow$	true/false
16E	like 15E, but the English word contains a dash, e.g., $Nobel - Prize \nearrow \frac{Nobel_{(\text{“}Nobel\text{”})}}{Preis_{(\text{“}prize\text{”})}} \searrow$	true/false
17E	like 15E, but also considering 1-to-n and n-to-1 links	true/false
18E	like 16E, but also considering 1-to-n and n-to-1 links	true/false

Table 2: List of new **source** language CRF features for compound merging.

"aufkommen" ("volume"), are both nouns and part of one common noun phrase, which is a strong indicator that the two words should be merged in German. In contrast, the syntactic relationship between "pay" (aligned to *"aufkommen"*) and "transport" (aligned to *"verkehr"*) is more distant[5]: merging is not indicated.

We also use the POS of the English words to learn (un)usual combinations of POS, independent of their exact syntactic structure (14E). Reconsider EXAMPLE (3): NN+NN is a more common POS pair for compounds than V+NN.

Finally, the alignment features (15E-18E) promote the merging into compounds whose alignments indicate that they should not have been split in the first place (e.g., *Rindfleisch*, 15E).

5.3 Compound Generation and Inflection

So far, we reported on how to decide **which** simple words are to be merged into compounds, but not **how** to recombine them. Recall from EXAMPLE (1) that the modifier of a compound sometimes needs to be transformed, before it can be combined with the head word (or next modifier), e.g., *"Ort"+"Zeit" = "Ortszeit"* ("local time").

We use SMOR to generate compounds from a combination of simple words. This allows us to create compounds with modifiers that never occurred as such in the training data. Imagine that *"Ort"* occurred only as compound head or as a single word in the training data. Using SMOR, we are still able to create the correct form of the modifier, including the required filler letter: *"Orts"*. This ability distinguishes our approach from previous approaches: Stymne and Cancedda (2011) do not reduce modifiers to their base forms[6] (they can only create new compounds when the modifier occurred as such in the training data) and Fraser et al. (2012) use a list for merging.

Finally, we use the system described in Fraser et al. (2012) to inflect the entire text.

6 Accuracy of Compound Prediction

We trained CRF models on the parallel training data (~40 million words)[7] of the EACL 2009 workshop on statistical machine translation[8] using different feature (sub)sets, cf. the "Experiment" column in Table 1 above. We examined the reliability of the CRF compound prediction models by applying them to held-out data:

1. split the German wmt2009 tuning data set
2. remember compound split points
3. predict merging with CRF models
4. combine predicted words into compounds
5. calculate f-scores on how properly the compounds were merged

Table 3 lists the CRF models we trained, together with their compound merging accuracies on held-out data. It can be seen that using more features (**SC→T→ST**) is favourable in terms of precision and overall accuracy and the positive impact of using source language features is clearer when only reduced feature sets are used (**TR** vs. **STR**).

However, these accuracies only somewhat correlate with SMT performance: while being trained and tested on clean, fluent German language, the

[5]Note that *"für etwas aufkommen"* (lit. "for sth. arise", idiom.: "to pay for sth.") is an idiomatic expression.

[6]They account for modifier transformations by using character n-gram features (cf.EXAMPLE (1)).

[7]However, target language feature frequencies are derived from the monolingual training data, ~146 million words.

[8]http://www.statmt.org/wmt09

exp	to be merged	all merged	correct merged	wrong merged	wrong not merged	merging wrong	precision	recall	f-score
SC	1,047	997	921	73	121	3	92.38%	88.13%	90.21%
T	1,047	979	916	59	128	4	93.56%	87.40%	90.38%
ST	1,047	976	917	55	126	4	93.95%	87.58%	90.66%
TR	1,047	893	836	52	204	5	93.62%	80.00%	86.27%
STR	1,047	930	866	58	172	6	93.12%	82.95%	87.74%

Table 3: Compound production accuracies of CRF models on held-out data: **SC**: re-implementation of Stymne and Cancedda (2011); **T**: all target language features, including a new one (cf. Table 1); **ST** = all **S**ource and **T**arget language features; **TR**: only a **r**educed set of target language features; **STR**: **TR**, plus all source language features given in Table 2.

exp	BLEU SCORES			#compounds found			
	mert.log	BLEU	RTS	all	ref	new	new*
RAW	14.88	14.25	1.0054	646	175	n.a.	n.a.
UNSPLIT	15.86	**14.74**	0.9964	661	185	n.a.	n.a.
SC	15.44	14.45	0.9870	882	241	47	8
T	15.56	14.32	0.9634	845	251	47	8
ST	15.33	14.51	0.9760	820	248	46	9
TR	15.24	14.26	0.9710	753	234	44	5
STR	15.37	**14.61**	0.9884	758	239	43	7
#compounds in reference text:				1,105	1,105	396	193

Table 4: SMT results. Tuning scores (mert.log) are on merged but uninflected data (except RAW). **RTS**: length ratio; **all**: #compounds produced; **ref**: reference matches; **new**: unknown to parallel data; **new***: unknown to target language data. **bold** face indicates statistical significance wrt. the RAW baseline, SC, T and TR.

models will later be applied to disfluent SMT output and might thus lead to different results there. Stymne and Cancedda (2011) dealt with this by *noisifying* the CRF training data: they translated the whole data set using an SMT system that was trained on the same data set. This way, the training data was less fluent than in its original format, but still of higher quality than SMT output of unseen data. In contrast, we left the training data as it was, but strongly reduced the feature set for CRF model training (e.g., no more use of surface words and POS tags, cf. **TR** and **STR** in Table 3) instead.

7 Translation Performance

We integrated our compound processing pipeline into an end-to-end SMT system. Models were trained with the default settings of the Moses SMT toolkit, v1.0 (Koehn et al., 2007) using the data from the EACL 2009 workshop on statistical machine translation. All compound processing systems are trained and tuned identically, except using different CRF models for compound prediction. All training data was split and reduced to the underspecified representation described in Section 4. We used KenLM (Heafield, 2011) with SRILM (Stolcke, 2002) to train a 5-gram language model based on all available target language training data. For tuning, we used batch-mira with '-safe-hope' (Cherry and Foster, 2012) and ran it separately for every experiment. We integrated the CRF-based merging of compounds into each iteration of tuning and scored each output with respect to an unsplit and lemmatised version of the tuning reference. Testing consists of:

1. translation into the split, underspecified German representation
2. compound merging using CRF models to predict recombination points
3. inflection of all words

7.1 SMT Results

We use 1,025 sentences for tuning and 1,026 sentences for testing. The results are given in Table 4. We calculate BLEU scores (Papineni et al., 2002) and compare our systems to a RAW baseline (built following the instructions of the shared task) and a baseline very similar to Fraser et al. (2012), using a lemmatised representation of words for decoding, re-inflecting them after translation, but without compound processing (UNSPLIT). Table 4 shows that only UNSPLIT and STR (source language and a reduced set of target language features) are significantly[9] improving over the RAW baseline. They also significantly outperform all other systems, except ST (full source and target language feature set). The difference between STR (14.61) and the UNSPLIT baseline (14.74) is **not** statistically significant.

[9]We used pair-wise bootstrap resampling with sample size 1000 and p-value 0.05, from: http://www.ark.cs.cmu.edu/MT

	group ID	example	reference	english	UNSPLIT	STR
lexically matches the reference	1a: perfect match	Inflationsrate	Inflationsrate	inflation rate	185	239
	1b: inflection wrong	Rohstoffpreisen	Rohstoffpreise	raw material prices	40	44
	2a: merging wrong	Anwaltsbewegung	Anwältebewegung	lawyers movement	5	9
	2b: no merging	Polizei Chef	Polizeichef	police chief	101	54
correct translation	3a: compound	Zentralbanken	Notenbank	central banks	92	171
	3b: no compound	pflanzliche Öle	Speiseöl	vegetable oils	345	291
wrong translation	4a: compound	Haushaltsdefizite	Staatshaushalts	state budget	12	42
	4b: no compound	Ansporn Linien	Nebenlinien	spur lines	325	255
Total number of compounds in reference text:					1,105	1,105

Table 5: Groups for detailed manual compound evaluation and results for **UNSPLIT** and **STR**.

reference	English source	UNSPLIT baseline		STR	
Teddybären	teddy bear	4b	Teddy tragen (Teddy, to bear)	1a	Teddybären (teddy bear)
Emissionsreduktion	emissions reduction	3b	Emissionen Reduzierung (emissions, reducing)	3a	Emissionsverringerung (emission decrease)
Geldstrafe	fine	4b	schönen (fine/nice)	3a	Bußgeld (monetary fine)
Tischtennis	table tennis	2b	Tisch Tennis (table, tennis)	4a	Spieltischtennis (play table tennis)
Kreditkartenmarkt	credit-card market	2b	Kreditkarte Markt (credit-card, market)	4a	Kreditmarkt (credit market)
Rotationstempo	rotation rate	2b	Tempo Rotation (rate, rotation)	4a	Temporotation (rate rotation)

Table 6: Examples of the detailed manual compound analysis for **UNSPLIT** and **STR**.

Compound processing leads to improvements at the level of unigrams and as BLEU is dominated by four-gram precision and length penalty, it does not adequately reflect compound related improvements. We thus calculated the number of compounds matching the reference for each experiment and verified whether these were known to the training data. The numbers in Table 4 show that all compound processing systems outperform both baselines in terms of finding more exact reference matches and also more compounds unknown to the training data. Note that STR finds less reference matches than e.g. T or ST, but it also produces less compounds overall, i.e. it is more precise when producing compounds.

However, as compounds that are correctly combined but poorly inflected are not counted, this is only a lower bound on true compounding performance. We thus performed two additional manual evaluations and show that the quality of the compounds (Section 7.2), and the human perception of translation quality is improving (Section 7.3).

7.2 Detailed Evaluation of Compounds

This evaluation focuses on how compounds in the the reference text have been translated.[10] We:

1. manually identify compounds in German reference text (1,105 found)
2. manually perform word alignment of these compounds to the English source text
3. project these English counterparts of compounds in the reference text to the decoded text using the "–print-alignment-info" flag
4. manually annotate the resulting tuples, using the categories given in Table 5

The results are given in the two rightmost columns of Table 5: besides a higher number of reference matches (cf. row 1a), STR overall produces more compounds than the UNSPLIT baseline, cf. rows 2a, 3a and 4a. Indirectly, this can also be seen from the low numbers of STR in category 2b), where the UNSPLIT baseline produces much more (101 vs. 54) translations that lexically match the reference without being a compound. While the 171 compounds of STR of category 3a) show that our system produces many compounds that are correct translations of the English, even though not matching the reference (and thus not credited by BLEU), the compounds of categories 2a) and 4a) contain examples where we either fail to reproduce the correct compound or over-generate compounds.

We give some examples in Table 6: for "teddy bear", the correct German word *"Teddybären"* is

[10] In another evaluation, we investigated the 519 compounds that our system produced but which did not match the reference: 367 were correct translations of the English, 87 contained erroneous lexemes and 65 were over-mergings.

missing in the parallel training data and instead of *"Bär"* ("bear"), the baseline selected *"tragen"* ("to bear"). Extracting all words containing the substring *"bär"* ("bear") from the original parallel training data and from its underspecified split version demonstrates that our approach is able to access all occurrences of the word. This leads to higher frequency counts and thus enhances the probabilities for correct translations. We can generalise over 18 different word types containing "bear" (e.g. "polar bears", "brown bears", "bear skin", "bear fur") to obtain only 2:

occurrences in raw training data: Bär (19), Bären (26), Bärendienst (42), Bärenfarmen (1), Bärenfell (2), Bärengalle(1), Bärenhaut (1), Bärenmarkt (1), Braunbär (1), Braunbären (3), Braunbärengebiete (1), Braunbär-Population (1), Eisbären(18), Eisbärenpopulation (2), Eisbärenpopulationen (1), Schwarzbär (1), Schwarzbären (1)

***"bär"* occurring in underspecified split data:**
Bär<+NN><Masc><Sg> (**94**)
Bär<+NN><Masc><Pl> (**29**)

"Emissionsverringerung" (cf. Table 6) is a typical example of group 3a): a correctly translated compound that does not lexically match the reference, but which is semantically very similar to the reference. The same applies for *"Bußgeld"*, a synonym of *"Geldstrafe"*, for which the UNSPLIT baseline selected *"schönen"* ("fine, nice") instead. Consider also the wrong compound productions, e.g. *"Tischtennis"* is combined with the verb *"spielen"* ("to play") into *"Spieltischtennis"*. In contrast, *"Kreditmarkt"* dropped the middle part *"Karte"* ("card"), and in the case of *"Temporotation"*, the head and modifier of the compound are switched.

7.3 Human perception of translation quality

We presented sentences of the UNSPLIT baseline and of STR in random order to two native speakers of German and asked them to rank the sentences according to preference. In order to prevent them from being biased towards compound-bearing sentences, we asked them to select sentences based on their native intuition, without revealing our focus on compound processing.

Sentences were selected based on source language sentence length: 10-15 words (178 sentences), of which either the reference or our system had to contain a compound (95 sentences). After removing duplicates, we ended up with 84 sentences to be annotated in two subsequent passes: first, without being given the reference sentence (approximating fluency), then, with the reference sentence (approximating adequacy). The results are given in Table 7. Both annotators preferred more sentences of our system overall, but the difference is clearer for the fluency task.

(a) Fluency: without reference sentence

$\kappa = 0.3631$		person 1			
		STR	UNSPLIT	equal	
person 2	STR	24	6	7	37
	UNSPLIT	5	16	9	30
	equal	6	2	9	17
		35	24	25	84

(b) Adequacy: with reference sentence

$\kappa = 0.4948$		person 1			
		STR	UNSPLIT	equal	
person 2	STR	23	4	5	32
	UNSPLIT	4	21	7	32
	equal	5	3	12	20
		32	28	24	84

Table 7: Human perception of translation quality.

8 Conclusion

Compounds require special attention in SMT, especially when translating into a compounding language. Compared with the baselines, all of our experiments that included compound processing produced not only many more compounds matching the reference exactly, but also many compounds that did not occur in the training data. Taking a closer look, we found that some of these new compounds could only be produced due to the underspecified representation we are using, which allows us to generalise over occurrences of simple words, compound modifiers and heads. Moreover, we demonstrated that features derived from the source language are a valuable source of information for compound prediction: experiments were significantly better compared with contrastive experiments without these features. Additional manual evaluations showed that compound processing leads to improved translations where the improvement is not captured by BLEU.

Acknowledgements

This work was supported by Deutsche Forschungsgemeinschaft grants Models of Morphosyntax for Statistical Machine Translation (Phase 2) and Distributional Approaches to Semantic Relatedness. We thank the anonymous reviewers for their comments and the annotators.

References

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *HLT-NAACL'12: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, volume 12, pages 34–35. Association for Computational Linguistics.

Chris Dyer. 2010. A Formal Model of Ambiguity and its Applications in Machine Translation. Phd dissertation, University of Maryland, USA.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling Inflection and Word Formation in SMT. In *EACL'12: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674. Association for Computational Linguistics.

Fabienne Fritzinger and Alexander Fraser. 2010. How to Avoid Burning Ducks: Combining Linguistic Analysis and Corpus Statistics for German Compound Processing. In *Proceedings of the Fifth Workshop on Statistical Machine Translation*, pages 224–234. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL '03: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–193, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demonstration Session*, pages 177–180. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML'01: Proceedings of the 18th International Conference on Machine Learning*.

Klaus Macherey, Andrew M. Dai, David Talbot, Ashok C. Popat, and Franz Och. 2011. Language-independent Compound Splitting with Morphological Operations. In *ACL '11: Proceedings of the 49th annual meeting of the Association for Computational Linguistics*, pages 1395–1404. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *ACL'02: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical Machine Translation of German Compound Words. In *FinTAL'06: Proceedings of the 5th International Conference on Natural Language Processing*, pages 616–624. Springer Verlag.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German Computational Morphology Covering Derivation, Composition and Inflection. In *LREC '04: Proceedings of the 4th Conference on Language Resources and Evaluation*, pages 1263–1266.

Andreas Stolcke. 2002. SRILM – an Extensible Language Modelling Toolkit. In *ICSLN'02: Proceedings of the international conference on spoken language processing*, pages 901–904.

Sara Stymne and Nicola Cancedda. 2011. Productive Generation of Compound Words in Statistical Machine Translation. In *EMNLP'11: Proceedings of the 6th Workshop on Statistical Machine Translation and Metrics MATR of the conference on Empirical Methods in Natural Language Processing*, pages 250–260. Association for Computational Linguistics.

Sara Stymne, Maria Holmqvist, and Lars Ahrenberg. 2008. Effects of Morphological Analysis in Translation between German and English. In *ACL'08: Proceedings of the 3rd workshop on statistical machine translation of the 46th annual meeting of the Association for Compuational Linguistics*, pages 135–138. Association for Computational Linguistics,.

Sara Stymne. 2009. A Comparison of Merging Strategies for Translation of German Compounds. In *EACL '09: Proceedings of the Student Research Workshop of the 12th conference of the European Chapter of the Association for Computational Linguistics*, pages 61–69. Association for Computational Linguistics.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying Morphology Generation Models to Machine Translation. In *ACL'08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 514–522. Association for Computational Linguistics.

Marion Weller, Alexander Fraser, and Sabine Schulte im Walde. 2013. Using Subcategorization Knowledge to Improve Case Prediction for Translation to German. In *ACL'13: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 593–603. Association for Computational Linguistics.