Target-side Word Segmentation Strategies for Neural Machine Translation

Matthias Huck, Simon Riess, Alexander Fraser

Center for Information and Language Processing
LMU Munich
Munich, Germany

{mhuck,fraser}@cis.lmu.de,riess.simon@campus.lmu.de

Abstract

For efficiency considerations, state-of-the-art neural machine translation (NMT) requires the vocabulary to be restricted to a limited-size set of several thousand symbols. This is highly problematic when translating into inflected or compounding languages. A typical remedy is the use of subword units, where words are segmented into smaller components. Byte pair encoding, a purely corpus-based approach, has proved effective recently.

In this paper, we investigate word segmentation strategies that incorporate more linguistic knowledge. We demonstrate that linguistically informed target word segmentation is better suited for NMT, leading to improved translation quality on the order of magnitude of $+0.5$ BLEU and -0.9 TER for a medium-scale English→German translation task.

Our work is important in that it shows that linguistic knowledge can be used to improve NMT results over results based only on the language-agnostic byte pair encoding vocabulary reduction technique.

1 Introduction

Inflection and nominal composition are morphological processes which exist in many natural languages. Machine translation into an inflected language or into a compounding language must be capable of generating words from a large vocabulary of valid word surface forms, or ideally even be open-vocabulary. In NMT, though, dealing with a very large number of target symbols is expensive in practice.

While, for instance, a standard dictionary of German, a compounding language, may cover 140 000 vocabulary entries,[1] NMT on off-the-shelf GPU hardware is nowadays typically only tractable with target vocabularies below 100 000 symbols.

This issue is made worse by the fact that compound words are not a closed set. More frequently occurring compound words may be covered in a standard dictionary (e.g., “Finanztransaktionssteuer”, English: “financial transaction tax”), but the compounding process allows for words to be freely joined to form new ones (e.g., “Finanztransaktionssteuerzahler”, English: “financial transaction tax payer”), and compounding is highly productive in a language like German.

Furthermore, a dictionary lists canonical word forms, many of which can have many inflected variants, with morphological variation depending on case, number, gender, tense, aspect, mood, and so on. The German language has four cases, three grammatical genders, and two numbers. German exhibits a rich amount of morphological word variations also in the verbal system. A machine translation system should ideally be able to produce any permissible compound word, and all inflections for each canonical form of all words (including compound words).

Previous work has drawn on byte pair encoding to obtain a fixed-sized vocabulary of subword units. In this paper, we investigate word segmentation strategies for NMT which are linguistically more informed. Specifically, we explore and empirically compare:

- Compound splitting.
- Suffix splitting.
- Prefix splitting.
- Byte pair encoding (BPE).
- Cascaded applications of the above.

[1] Duden, 26th ed., 2013, cf. http://www.duden.de/ueber_duden/auflagengeschichte.

Proceedings of the Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 56–67
Copenhagen, Denmark, September 711, 2017. ©2017 Association for Computational Linguistics

Our empirical evaluation focuses on target-language side segmentation, with English→German translation as the application task. Our proposed approaches improve machine translation quality by up to $+0.5$ BLEU and -0.9 TER, respectively, compared with using plain BPE.

Advantages of linguistically-informed target word segmentation in NMT are:

1. *Better vocabulary reduction* for practical tractability of NMT, as motivated above.
2. *Reduction of data sparsity*. Learning lexical choice is more difficult for rare words that appear in few training samples (e.g., rare compounds), or when a single form from a source language with little inflection (such as English) has many target-side translation options which are morphological variants. Splitting compounds and separating affixes from stems can ease lexical selection.
3. *Better open vocabulary translation*. With target-side word segmentation, the NMT system can generate sequences of word pieces at test time that have not been seen in this combination in training. It may produce new compounds, or valid morphological variants that were not present in the training corpus, e.g. by piecing together a stem with an inflectional suffix in a new, but linguistically admissible way. Using a linguistically informed segmentation should better allow the system to try to learn the linguistic processes of word formation.

2 Word Segmentation Strategies

2.1 Byte Pair Encoding

A technique in the manner of the Byte Pair Encoding (BPE) compression algorithm (Gage, 1994) can be adopted in order to segment words into smaller subword units, as suggested by Sennrich et al. (2016b). The BPE word segmenter conceptionally proceeds by first splitting all words in the whole corpus into individual characters. The most frequent adjacent pairs of symbols are then consecutively merged, until a specified limit of merge operations has been reached. Merge operations are not applied across word boundaries. The merge operations learned on a training corpus can be stored and applied to other data, such as test sets.

suffixes
-e, -em, -en, -end, -enheit, -enlich, -er, -erheit, -erlich, -ern, -es, -est, -heit, -ig, -igend, -igkeit, -igung, -ik, -isch, -keit, -lich, -lichkeit, -s, -se, -sen, -ses, -st, -ung
prefixes
ab-, an-, anti-, auf-, aus-, auseinander-, außer-, be-, bei-, binnen-, bitter-, blut-, brand-, dar-, des-, dis-, durch-, ein-, empor-, endo-, ent-, entgegen-, entlang-, entzwei-, epi-, er-, extra-, fehl-, fern-, fest-, fort-, frei-, für-, ge-, gegen-, gegenüber-, grund-, heim-, her-, hetero-, hin-, hinter-, hinterher-, hoch-, homo-, homöo-, hyper-, hypo-, inter-, intra-, iso-, kreuz-, los-, miss-, mit-, mono-, multi-, nach-, neben-, nieder-, non-, pan-, para-, peri-, poly-, post-, pro-, prä-, pseudo-, quasi-, schein-, semi-, stock-, sub-, super-, supra-, tief-, tod-, trans-, ultra-, um-, un-, unab-, unan-, unauf-, unaus-, unbe-, unbei-, undar-, undis-, undurch-, unein-, unent-, uner-, unfehl-, unfort-, unfrei-, unge-, unher-, unhin-, unhinter-, unhoch-, unmiss-, unmit-, unnach-, unter-, untief-, unum-, ununter-, unver-, unvor-, unweg-, unwider-, unzer-, unzu-, unüber-, ur-, ver-, voll-, vor-, voran-, voraus-, vorüber-, weg-, weiter-, wider-, wieder-, zer-, zu-, zurecht-, zurück-, zusammen-, zuwider-, über-

Table 1: German affixes which our suffix splitter and prefix splitter separate from the word stem.

An advantage of BPE word segmentation is that it allows for a reduction of the amount of distinct symbols to a desired order of magnitude. The technique is purely frequency-based. Frequent sequences of characters will be joined through the merge operations, resulting in common words not being segmented. Words containing rare combinations of characters will not be fully merged from the character splitting all the way back to their original form. They will remain split into two or more subword units in the BPE-segmented data. On the downside, the BPE algorithm has no notion of morphosyntax, narrowing down its capabilities at modeling inflection and compounding. BPE also has no guidelines for splitting words into syllables. This way no phonetic or semantic substructures are taken into account. Therefore BPE splits often appear arbitrary to the human reader, since it appears frequently that subword units ignore syllable boundaries entirely.

Nevertheless, NMT systems incorporating BPE word segmentation have achieved top translation quality in recent shared tasks (Sennrich et al., 2016a; Bojar et al., 2016). We designed our linguistically-informed segmentation techniques by looking at the shortcomings of BPE segmentations.

2.2 Compound Splitting

BPE word segmentation operates bottom-up from characters to larger units. Koehn and Knight (2003) have proposed a frequency-based word segmentation method that starts from the other end, top-down inspecting full words and looking into whether they are composed of parts which are proper words themselves. Any composed word is segmented into parts such that the geometric mean of word frequencies of its parts (counted in the original corpus) is maximized. This technique represents a suitable approach for compound splitting in natural language processing applications. It has been successfully applied in numerous statistical machine translation systems, mostly on the source language side, but sometimes also on the target side (Sennrich et al., 2015).

The difference in nature between BPE word segmentation and frequency-based compound splitting (bottom-up and top-down) leads to quite different results. While BPE tends to generate unintuitive splits, compound splitting nearly always comes up with reasonable word splits. On the other hand there are many possible intuitive word splits that compound splitting does not catch.

2.3 Suffix Splitting

Morphological variation in natural languages is often realized to a large extent through affixation. In the German language there are several suffixes that unambiguously mark a word as an adjective, noun, or verb. By splitting these telling suffixes, we can automatically include syntactic information. Even though this underlying relationship between suffix and morphological function is sometimes ambiguous—especially for verbs—reasonable guesses about the POS of a word with which we are not familiar are only possible by considering its suffix.

Information retrieval systems take advantage of this observation and reduce search queries to stemmed forms by means of simply removing common suffixes, prefixes, or both. The Porter stemming algorithm is a well-known affix stripping method (Porter, 1980). In such algorithms, some basic linguistic knowledge about the morphology of a particular language is taken into account in order to come up with a few handwritten rules which would detect common affixes and delete them. We can benefit from the same idea for the segmentation of word surface forms.

We have modified the Python implementation of the German Snowball stemming algorithm from NLTK[2] for our purposes. The Snowball stemmer removes German suffixes via some language-specific heuristics. In order to obtain a segmenter, we have altered the code to not drop suffixes, but to write them out separately from the stem. Our Snowball segmenter splits off the German suffixes that are shown in Table 1. Some of them are inflectional, others are used for nominalization or adjectivization. The suffix segmenter also splits sequential appearances of suffixes into multiple parts according to the Snowball algorithm's splitting steps, but always retaining a stem with a minimum length of at least three characters.

Table 2 shows some relationships between German suffixes and their English translations. Especially nominalizations and participles are particularly consistent, which makes translation rather unambiguous. Even though an exact translation from every German suffix to one specific English suffix cannot be established, this shows that a set of German suffixes translates into a set of English suffixes. Some suffixes indeed have an unambiguous translation like German *-los* to English *-less* or German *-end* to English *-ing*. These relationships might be due to the shared roots of the German and English language. Especially for other Germanic languages this promises transferability of our results.

It seems to be a reasonable assumption that other languages also have a certain set of possible suffixes which correspond to each type of word. For these relationships our approach may be able to automatically and cheaply add (weak) POS information, which might improve translation quality, but this will require further investigation in future work.

We would also like to study the relationship between stemming quality and resulting NMT translation quality. Weissweiler and Fraser (2017) have introduced a new stemmer of German and showed that it performs better than Snowball using comparison with gold standards. This may serve as an interesting starting point.

2.4 Prefix Splitting

Similarly to our Snowball suffix segmenter, we have written a small script to split off prefixes.

[2] http://www.nltk.org/_modules/nltk/stem/snowball.html

German suffixes unambiguously marking nouns
-ung, *-heit*, *-nis*, *-keit*, *-sal*, *-schaft*, *-ling*, *-tum*
English nominalizations with *-ness* are translated consistently by adding one of these suffixes
busyness – Geschäftigkeit
abstractness – Abstraktheit
kindness – Freundlichkeit
coziness – Behaglichkeit
giftedness – Begabung
sadness – Traurigkeit
tiredness – Müdigkeit
laziness – Faulheit
But a simple mapping between German and English noun suffixes does not exist
Abholzung – deforestation
Segmentierung – segmentation
Trockenheit – aridity
Obrigkeit – autority
Genauigkeit – precision
Bündnis – alliance
Gefängnis – prison
Verhältnis – relationship
German suffixes typical for adjectives
-ig, *-lig*, *-isch*, *-sam*, *-bar*, *-haft*, *-los*
Adjective derivation using these suffixes
achtsam – mindful
wendig – agile
begehbar – accessible
sichtbar – visible
nahrhaft – nutricious
essbar – edible
fettig – greasy
ethisch – ethical
moralisch – morally
laienhaft – unprofessional
-los with consistent English counterpart *-less*
taktlos – tactless
reglos – motionless
rastlos – restless
schamlos – shameless
German participles ending with *-end*
hängend – hanging
stehend – standing
schlafend – sleeping
lachend – laughing

Table 2: Examples illustrating the use of German suffixes.

The common German verb prefix *ver-* shows no obvious pattern in English translations
verstehen – to understand
sich verirren – to get lost
vergehen – to vanish
sich versprechen – to misspeak oneself
verfehlen – to miss
aus Versehen – unintentionally
verbieten – to prohibit
vergessen – to forget
Another common German verb prefix, *be-*, also shows no obvious pattern
behaupten – to claim
beschuldigen – to accuse
bewerben – to apply for
beladen – to load
betonen – to emphasize
bewahren – to preserve
The common German prefix *auf-* (English: *on*, *up*) has relatively consistent pattern in English translation
aufstellen – to put up
aufsetzen – to sit up
aufgehen – to give up
aufstehen – to stand up
aufblasen – to blow up
aufgeben – to give up
aufbauen – to set up
aufhören – to stop
German verb *setzen* (English: *to sit down*) with different prefixes
absetzen – to drop off
besetzen – to occupy
ersetzen – to replace
zersetzen – to decompose
umsetzen – to realize
widersetzen – to defy

Table 3: Examples illustrating the use of German prefixes.

Here, we specifically target verb and adjective prefixes and thus only segment lowercase words, excluding nouns which are written in uppercase in German text. We consider the prefixes as shown in Table 1. We sort them descending by length, checking for longer prefix matches first. Negational prefixes (beginning with *un-*, but not *unter-*) are additionally segmented after *un-*; e.g., *unab-* becomes *un- ab-*. In case the remaining part starts with either of the two verb infixes *-zu-* or *-ge-*, we also segment after that infix. We require the final stem to be at least three characters long.

While suffixes tend to contain morphological information, German prefixes change—sometimes radically—the semantics of the word stem. Some prefixes, especially those indicating local relationships, have a relatively clear and consistent translation. On the other hand, certain prefixes change the meaning more subtly and also more ambiguously. Therefore some prefixes lead to a simple translation while others change the meaning too radically.

Table 3 shows how the meaning of German verbs can change by adding different prefixes to a common stem. The example for *setzen – to sit down* illustrates that each of the combinations is semantically so different from the others that their translations have to be learned separately. This also means that splitting the prefix might not benefit the machine translation system, since generalization is hardly possible.

The examples given in Table 3 also suggest that a single verb prefix may affect the semantics of the word in ambiguous ways when applied to different verb stems. The very common German prefix *ver-*, for instance, which often indicates an incorrectly performed action (like *sich versprechen – to misspeak oneself* or *verfehlen – to miss*), still has a lot of different applications. This variety shows that prefixes clearly carry information, but still it is highly ambiguous and therefore might not benefit the translation process.

The German prefix *auf – up, on* has a relatively unambiguous translation, though, and hence splitting it might support the machine translation system. A possible improvement might be only splitting these unambiguously translatable prefixes (which in general are prepositions indicating the direction of the altered verb), but this remains to be investigated in future research.

2.5 Cascaded Application of Segmenters

Affix splitting and compound splitting can be applied in combination, by cascading the segmenters and preprocessing the data first with the suffix splitter, then optionally with the prefix splitter, and then with the compound splitter. In a cascaded application, the compound splitter is applied to word stems only, and the counts for computing the geometric means of word frequencies for compound splitting are collected after affix splitting.

When cascading the compound splitter with affix splitting, we introduce a minor modification. Our standalone compound splitter takes the filler letter "*s*" and "*es*" into account, which often appear in between word parts in German noun compounding. For better consistency of the compound splitting component with affix splitting, we additionally allow for more fillers, namely: suffixes, suffixes followed by "*s*", and "*zu*".

The methods for compound splitting, suffix splitting, and prefix splitting provide linguistically more sound approaches for word segmentation, but they do not arbitrarily reduce the amount of distinct symbols. For a further reduction of the number of target-side symbols, we may want to apply a final BPE segmentation step on top of the other segmenters. BPE will not re-merge words that have been segmented before. It can benefit from the prior segmentation provided to it and come up with a potentially better sequence of merge operations. Affixes will be learned as subwords but not joined with the stem. This improves the quality of resulting BPE splits. BPE no longer combines arbitrary second to last syllables with their suffixes, which makes learning the other—non affix—syllables easier.

We deliberately decided against joint/bilingual BPE, for multiple reasons. (1.) In cascaded segmentations, BPE operations are learned from training data after previous splitters in the pipeline have been applied. With joint BPE, the source would be affected, being preprocessed slightly differently in different setups. Instead, we opted for conducting BPE-50K separately over English. The source is hence equal in all setups, which we believe renders the evaluation more sound. (2.) When tying source+target in joint-BPE, vocabulary sizes cannot be controlled independently on each side. Joint-BPE with 59500 operations for instance yields 46K German types in the data, but an English corpus containing only 26K types.

BPE	*sie alle versch ## icken vorsätzlich irreführende Dokumente an Kleinunternehmen in ganz Europa .*
compound + BPE	*sie alle verschicken vorsätzlich #L irre @@ führende Dokumente an #U klein @@ unter @@ nehmen in ganz Europa .*
suffix + BPE	*sie all $$e verschick $$en vorsätz $$lich irreführ $$end $$e Dokument $$e an Kleinunternehm $$en in ganz Europa .*
suffix + compound + BPE	*sie all $$e verschick $$en vorsätz $$lich #L Irre @@ führ $$end $$e Dokument $$e an #U klein @@ Unternehm $$en in ganz Europa .*
suffix + prefix + compound + BPE	*sie all $$e ver§§ schick $$en vor§§ sätz $$lich #L Irre @@ führ $$end $$e Dokument $$e an #U klein @@ Unternehm $$en in ganz Europa .*
English	*they all mail deliberately deceptive documents to small businesses across Europe .*

Table 4: Different word segmentation strategies applied to a training sentence. ## is a BPE split-point, ver§§ is prefix *ver*, $$en is the suffix *en*, #U and #L are upper and lower case indicators for compounds, @@ indicates a compound merge-point, @s@ would indicate a compound merged with the letter *s* between the parts, etc.

(3.) Joint-BPE may boost transliteration capabilities. Generally, we would however recommend to extract BPE operations monolingually to better capture the properties of the individual language. We argue that well justified segmentation cannot be language-independent. (4.) We would not expect fundamentally different findings when switching to joint-BPE everywhere.

2.6 Reversibility

Target-side word segmentation needs to be reversible in postprocessing. We introduce special markers to enable reversibility of word splits. For suffixes, we attach a marker to the beginning of each suffix token; for prefixes to the end of each split prefix.

Fillers within segmented compounds receive attached markers on either side. When a compound is segmented into parts with no filler between them, we place a separate special marker token in the middle which is not attached to any of the parts. It indicates the segmentation and has two advantages over attaching it to any of the parts: (1.) The tokens of the parts are exactly the same as when they appear as words outside of a compound. The NMT system does not perceive them as different symbols. (2.) There is more flexibility at producing new compounds that have not been seen in the training corpus. The NMT system can decide to place any symbol into a token sequence that would form a compound, even the ones which were never part of a compound in training. The vocabulary is more open in that respect.

We adhere to the same rationale for split markers in BPE word segmentation. A special marker token is placed separately between subword units, with whitespace around it. In our experience, attaching the marker to BPE subword units does not improve translation quality over our practice.

The compound splitter alters the casing of compound parts to the variants that appears most frequently in the corpus. When merging compounds in postprocessing, we need to know whether to lowercase or to uppercase the compound. We let the translation system decide and introduce another special annotation in order to allow for this. When we segment compounds, we always place an indicator symbol before the initial part of the split compound token sequence, which can be either *#L* or *#U*. It specifies the original casing of the compound (lower or upper).

The effect of different segmentation strategies on the word splits in an example sentence is shown in Table 4.

Preprocessing	**#types**	**#tokens**
tokenized	303 K	39 M
compound	139 K	45 M
suffix	217 K	54 M
suffix + compound	98 K	60 M
suffix + prefix + compound	88 K	63 M
BPE	46 K	42 M
compound + BPE	46 K	46 M
suffix + BPE	45 K	56 M
suffix + compound + BPE	43 K	60 M
suffix + prefix + compound + BPE	43 K	64 M

Table 5: Target-side training corpus statistics.

System	**test2007**		**test2008**	
	BLEU	TER	BLEU	TER
top 50K voc. (source & target)	25.5	60.9	25.2	60.9
BPE	25.8	60.7	25.6	60.9
compound + BPE	25.9	60.3	25.5	60.6
suffix + BPE	**26.3**	60.0	**26.0**	**60.1**
suffix + compound + BPE	26.2	**59.8**	25.8	60.2
suffix + prefix + compound + BPE	26.1	**59.8**	25.9	60.6
suffix + prefix + compound, 50K	25.9	59.9	25.5	60.3
phrase-based (Huck et al., 2015)	22.6	–	22.1	–

Table 6: English→German experimental results on Europarl (case-sensitive BLEU and TER).

3 Machine Translation Experiments

3.1 Experimental Setup

We conduct an empirical evaluation using encoder-decoder NMT with attention and gated recurrent units as implemented in Nematus (Sennrich et al., 2017). We train and test on English–German Europarl data (Koehn, 2005). The data is tokenized and frequent-cased using scripts from the Moses toolkit (Koehn et al., 2007). Sentences with length >50 after tokenization are excluded from the training corpus, all other sentences (1.7 M) are considered in training under every word segmentation scheme. We set the amount of merge operations for BPE to 50K. Corpus statistics of the German data after different preprocessings are given in Table 5. On the English source side, we apply BPE separately, also with 50K merge operations.

For comparison, we build a setup denoted as *top 50K voc. (source & target)* where we train on the tokenized corpus without any segmentation, limiting the vocabulary to the 50K most frequent words on each side and replacing rare words by "*UNK*". In a setup denoted as *suffix + prefix + compound, 50K*, we furthermore examine whether BPE can be omitted in a cascaded application of target word segmenters. Here, we use the top 50K target symbols after suffix, prefix, and compound splitting, but still apply BPE to the English source.

It is important to note that the amount of distinct target symbols in the setups ranges between 43K-46K; 50K for top-50K-voc systems. There are no massive vocabulary size differences. We always apply 50K BPE operations. Minor divergences in the number of types naturally occur amongst the various cascaded segmentations. The linguistically-informed splitters segment more, resulting in more tokens. We chose BPE-50K because the vocabulary is reasonably large while training fits onto GPUs with 8 GB of RAM. Larger vocabularies come at the cost of either more RAM or adjustment of other parameters (e.g., batch size or sentence length limit). From hyperparameter search over reduced vocabulary sizes we would not expect important insights, so we do not do this.

In all setups the training samples are always the same. We removed long sentences after tokenization but before segmentation, which affects all setups equally. No sentences are discarded after that stage (Nematus' maxlen > longest sequence in data).

We configure dimensions of 500 for the embeddings and 1024 for the hidden layer. We train with the Adam optimizer (Kingma and Ba, 2015), a learning rate of 0.0001, batch size of 50, and dropout with probability 0.2 applied to the hidden layer.[3] We validate on the *test2006* set after every 10 000 updates and do early stopping when the validation cost has not decreased for ten epochs.

We evaluate case-sensitive with BLEU (Papineni et al., 2002) and TER (Snover et al., 2006), computed over postprocessed hypotheses against the raw references with `mteval-v13a` and `tercom.7.25`, respectively.

3.2 Experimental Results

The translation results are reported in Table 6. Cascading compound splitting and BPE slightly improves translation quality as measured in TER. Cascading suffix splitting with BPE or with compound splitting plus BPE considerably improves translation quality by up to +0.5 BLEU or −0.9 TER over pure BPE. Adding in prefix splitting is less effective. We conjecture that prefix

[3] In preliminary experiments, we found dropout for source, target, and embeddings did not yield additional gains.

Words in output System	BPE-merged tokens	 types	compound-merged tokens	 types	suffix-merged tokens	 types	prefix-merged tokens	 types
BPE	1 075 (1.9 %)	1 032 (13.4 %)	– –	– –	– –	– –	– –	– –
compound + BPE	271 (0.5 %)	255 (3.3 %)	2 766 (4.9 %)	1 738 (22.6 %)	– –	– –	– –	– –
suffix + BPE	443 (0.8 %)	427 (5.6 %)	– –	– –	19 152 (33.7 %)	4 915 (64.0 %)	– –	– –
suffix + compound + BPE	111 (0.2 %)	106 (1.4 %)	2 568 (4.5 %)	1 597 (20.4 %)	19 028 (33.7 %)	5 022 (64.1 %)	– –	– –
suffix + prefix + compound + BPE	100 (0.2 %)	95 (1.2 %)	2 566 (4.5 %)	1 577 (20.2 %)	19 063 (33.5 %)	4 990 (64.0 %)	4 601 (8.1 %)	1 667 (21.4 %)

Table 7: Statistics over words in system outputs for *test2008*, after desegmentation.

Words in output System	overall tokens	 types	 ratio
BPE	57 334	7 700	0.134
compound + BPE	56 827	7 692	0.135
suffix + BPE	56 849	7 674	0.135
suffix + compound + BPE	56 461	7 839	0.139
suffix + prefix + compound + BPE	56 875	7 797	0.137
reference	57 073	8 975	0.157

Table 8: Overall types and tokens, measured on *test2008* after desegmentation (hypotheses translations) or after tokenization (reference).

System	avg. sent. length
BPE	28.7
compound + BPE	28.4
suffix + BPE	28.4
suffix + compound + BPE	28.2
suffix + prefix + compound + BPE	28.4
reference	28.5

Table 9: Average sentence lengths on *test2008*.

Words in output System	unseen vocabulary tokens	 types
BPE	197 (0.3 %)	194 (2.5 %)
compound + BPE	280 (0.5 %)	257 (3.3 %)
suffix + BPE	139 (0.2 %)	138 (1.8 %)
suffix + compound + BPE	262 (0.5 %)	238 (3.0 %)
suffix + prefix + compound + BPE	265 (0.5 %)	234 (3.0 %)

Table 10: Productivity at open vocabulary translation, measured on *test2008* system outputs (after desegmentation) against the vocabulary of the tokenized training data.

splitting does not help because German verb prefixes often radically modify the meaning. When prefixes are split off, the decoder's embeddings layer may therefore become less effective (as the stem may be confusable with a completely different word).

We also evaluated casing manually. Manual inspection of the first fifty *#L* / *#U* occurrences in one of the hyptheses reveals that none is misplaced, and casing is always correctly indicated.

3.3 Analysis

In order to better understand the impact of the different target-side segmentation strategies, we analyze and compare the output of our main setups. Particularly, we turn our attention on the words in the translation outputs for the *test2008* set. For the analysis, in order to achieve comparable vocabularies in the various outputs, we apply desegmentation to all of the plain hypotheses produced by the systems. However, we do not run the full postprocessing pipeline: detruecasing and detokenization are omitted.

First, we count the number of words in the desegmented translations that have been merged together from subword components in the plain system outputs. Table 7 shows the statistics. The table rows contain the absolute amounts and relative frequencies of words with subword unit parts in the desegmented hypotheses, for running words in the text (types) and in terms of the vocabulary in the *test2008* translation output. The frequencies are relative to all words in the respective output. Note that when cascaded word segmentation was applied, a single desegmented word may be composed of multiple subword units that originate from different word splitters. We find that compared to the pure BPE system, many more words

OOV types	A	B	C	D	E
A	0	1621 (21.1 %)	1583 (20.6 %)	1584 (20.6 %)	1626 (21.1 %)
B	1612 (21.0 %)	0	1589 (20.7 %)	1469 (19.1 %)	1434 (18.7 %)
C	1559 (20.3 %)	1574 (20.5 %)	0	1451 (18.9 %)	1456 (19.0 %)
D	1726 (22.0 %)	1620 (20.7 %)	1617 (20.6 %)	0	1435 (18.3 %)
E	1725 (22.1 %)	1542 (19.8 %)	1579 (20.3 %)	1392 (17.9 %)	0
R	3641 (40.6 %)	3676 (41.0 %)	3624 (40.4 %)	3604 (40.2 %)	3634 (40.5 %)

Table 11: Systems compared against each other in terms of types found in *test2008* hypothesis translations, after desegmentation. (OOV words of output of vertical system wrt. vocabulary present in output of horizontal system.) *A*: BPE. *B*: compound + BPE. *C*: suffix + BPE. *D*: suffix + compound + BPE. *E*: suffix + prefix + compound + BPE. *R*: reference translation.

OOV tokens	A	B	C	D	E
A	0	1804 (3.1 %)	1763 (3.1 %)	1801 (3.1 %)	1826 (3.2 %)
B	1814 (3.2 %)	0	1793 (3.2 %)	1663 (2.9 %)	1612 (2.8 %)
C	1741 (3.1 %)	1768 (3.1 %)	0	1647 (2.9 %)	1648 (2.9 %)
D	1942 (3.4 %)	1803 (3.2 %)	1801 (3.2 %)	0	1565 (2.8 %)
E	1958 (3.4 %)	1734 (3.0 %)	1794 (3.2 %)	1554 (2.7 %)	0
R	4506 (7.9 %)	4582 (8.0 %)	4484 (7.9 %)	4484 (7.9 %)	4520 (7.9 %)

Table 12: Systems compared against each other in terms of tokens found in *test2008* hypothesis translations, after desegmentation.

Output similarity	A	B	C	D	E
A	100	61.6	61.3	60.4	60.1
B	61.6	100	61.4	62.0	62.1
C	61.3	61.4	100	62.5	62.9
D	60.5	62.0	62.5	100	63.0
E	60.1	62.1	62.9	63.0	100

Table 13: System outputs (after desegmentation) evaluated against each other with BLEU. (Hypothesis translation of vertical system against output of horizontal system as the reference in `multi-bleu.perl`.)

are merged from subword unit parts in the other systems.

Table 8 presents the overall amount of types and tokens in the hypothesis translations and in the reference. The pure BPE system exhibits the lowest type/token ratio, whereas the type/token ratio in the reference is higher than in all the machine translation outputs.

Average sentence lengths are given in Table 9. The pure BPE system produces sentences that are slightly longer than the ones in the reference. All other setups tend to be below the average reference sentence length, the shortest sentences being produced by the *suffix* + *compound* + *BPE* system.

Next, we look into how often the open vocabulary capabilities of the systems lead to the generation of words which are not present in the tokenized training corpus. We denote these words as “unseen”. Table 10 reveals that only small fractions of the words formed from subword unit parts (as counted before, Table 7) are unseen. The relative frequency of produced unseen words is smaller than—or equal to—half a percent in the running text. The setups trained with compound-split target data produce unseen words a bit more often. While at first glance it might seem disappointing that the systems' open vocabulary capabilities do not come into effect more heavily, this observation however emphasizes that we have succeeded at training neural models that adhere to word formation processes which lead to valid forms.

A straightforward follow-up question is how lexically dissimilar the various system outputs are. In Tables 11 and 12, we compare all hypotheses pairwise against each other, measuring the amount of words in one hypothesis that does not appear in the vocabulary present in a translation from another system. We basically calculate cross-hypothesis out-of-vocabulary (OOV) rates. Table 11 shows the results on type level, Table 12 on token level. We furthermore compare against the reference. The system outputs are lexically quite dissimilar, but much closer to each other than to the reference.

We can finally follow the very same rationale by evaluating the system outputs against each other with BLEU, calculating the BLEU score of one hypothesis against another hypothesis rather than against a reference translation. The result, presented in Table 13, reaffirms that the different systems have each learned to translate in different ways, based on the respective segmentation of the training data.

Our cascaded *suffix* + *compound* + *BPE* target word segmentation strategy was employed for LMU Munich's participation in the WMT17 shared tasks on machine translation of news and of biomedical texts. We refer the reader to the system description paper (Huck et al., 2017a), where we include some interesting translation examples from the news translation task. We note that our system was ranked first in the human evaluation of the news task, despite having a lower BLEU score than Edinburgh's submission. BLEU, which tries to automatically predict how humans will evaluate quality, may unfairly penalize approaches like ours, but more study is needed.

4 Related Work

The SMT literature has a wide diversity of approaches in dealing with translation to morphologically rich languages. One common theme is modeling the relationship between lemmas and surface forms using morphological knowledge, e.g., (Toutanova and Suzuki, 2007; Koehn and Hoang, 2007; Bojar and Kos, 2010; Fraser et al., 2012; Weller et al., 2013; Tamchyna et al., 2016; Huck et al., 2017b). This problem has been studied for NMT by Tamchyna et al. (2017), and it would be interesting to compare with their approach.

Our work is closer in spirit to previous work on integrating morphological segmentation into SMT. Some examples of early work here include work on Arabic (Lee et al., 2003) and Czech (Goldwater and McClosky, 2005). More recent work includes work on Arabic, such as (Habash, 2007), and work on Turkish (Oflazer and Durgar El-Kahlout, 2007; Yeniterzi and Oflazer, 2010). Unsupervised morphological splitting, using, e.g., Morfessor has also been tried, particularly for dealing with agglutinative languages (Virpioja et al., 2007). Our work is motivated by the same linguistic observations as theirs.

Other studies, e.g., (Popović et al., 2006; Stymne, 2008; Cap et al., 2014), model German compounds by splitting them into single simple words in the SMT training data, and then predicting where to merge simple words as a post-processing step (after SMT decoding). This has similarities to our use of compound splitting and markers in NMT.

There is also starting to be interest in alternatives to BPE in NMT. The Google NMT system (Wu et al., 2016) used wordpiece splitting, which is similar to but different from BPE and would be interesting to evaluate in future work. Ataman et al. (2017) considered both supervised and unsupervised splitting of agglutinative morphemes in Turkish, which is closely related to our ideas. An important difference here is that Turkish is an agglutinative language, while German has fusional inflection and very productive compounding.

We are also excited about early work on character-based NMT such as (Lee et al., 2016), which may eventually replace segmentation models like those in our work (or also replace BPE when linguistically aware segmentation is not available). However, at the current stage of research character-based approaches require very long training times and extensive optimization of hyperparameters to make them work, and still do not seem to be able to produce state-of-the-art translation quality on a wide range of tasks. More research is needed in making character-based NMT robust and accessible to many research groups.

5 Conclusion

Linguistically motivated target-side word segmentation improves neural machine translation into an inflected and compounding language. The system can learn linguistic word formation processes from the segmented data. For German, we have shown that cascading of suffix splitting—or suffix splitting and compound splitting—with BPE yields the best results. In future work we will consider alternative sources of linguistic knowledge about morphological processes and also evaluate high performance unsupervised segmentation.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 644402 (HimL). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. In *Proceedings of EAMT*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar and Kamil Kos. 2010. 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden. Association for Computational Linguistics.

Fabienne Cap, Alexander Fraser, Marion Weller, and Aoife Cahill. 2014. How to produce unseen teddy bears: Improved morphological processing of compounds in SMT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 579–587.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 664–674, Avignon, France.

Philip Gage. 1994. A New Algorithm for Data Compression. *C Users J.*, 12(2):23–38.

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, Vancouver, Canada.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *MTSUMMIT*, Copenhagen, Denmark.

Matthias Huck, Alexandra Birch, and Barry Haddow. 2015. Mixed-Domain vs. Multi-Domain Statistical Machine Translation. In *Proc. of MT Summit XV, vol.1: MT Researchers' Track*, pages 240–255, Miami, FL, USA.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017a. LMU Munich's Neural Machine Translation Systems for News Articles and Health Information Texts. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Aleš Tamchyna, Ondřej Bojar, and Alexander Fraser. 2017b. Producing Unseen Morphological Variants in Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 369–375, Valencia, Spain. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit X*, Phuket, Thailand.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning (EMNLP-CONLL)*, Prague, Czech Republic.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–194, Budapest, Hungary. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *CoRR*, abs/1610.03017.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. 2003. Language model based Arabic word segmentation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan.

Kemal Oflazer and Ilknur Durgar El-Kahlout. 2007. Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic. 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 *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of German compound words. In *FinTAL - 5th International Conference on Natural Language Processing, Springer Verlag, LNCS*, pages 616–624, Turku, Finland.

Martin Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Philip Williams, and Matthias Huck. 2015. A tree does not make a well-formed sentence: Improving syntactic string-to-tree statistical machine translation with more linguistic knowledge. *Computer Speech & Language*, 32(1):27–45.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231, Cambridge, MA, USA.

Sara Stymne. 2008. German compounds in factored statistical machine translation. In *GoTAL 2008: Proceedings of the 6th International Conference on Natural Language Processing*, pages 464–475. Springer Verlag.

Aleš Tamchyna, Alexander Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. 2016. Target-Side Context for Discriminative Models in Statistical Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1704–1714, Berlin, Germany. Association for Computational Linguistics.

Aleš Tamchyna, Marion Weller-Di Marco, and Alexander Fraser. 2017. Target-Side Inflection in Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Kristina Toutanova and Hisami Suzuki. 2007. Generating case markers in machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies Conference (NAACL-HLT)*, pages 49–56, Rochester, NY.

Sami Virpioja, Jaakko J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *PROC. OF MT SUMMIT XI*, pages 491–498.

Leonie Weissweiler and Alexander Fraser. 2017. Developing a stemmer for German based on a comparative analysis of publicly available stemmers. In *Proceedings of the German Society for Computational Linguistics and Language Technology (GSCL)*, Berlin, Germany.

Marion Weller, Alexander Fraser, and Sabine Schulte im Walde. 2013. Using subcategorization knowledge to improve case prediction for translation to German. In *Proceedings of the 51st Annual Conference of the Association for Computational Linguistics (ACL)*, pages 593–603, Sofia, Bulgaria.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *Proceedings of the 48th Annual Conference of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden. Association for Computational Linguistics.