Baltic J. Modern Computing, Vol. 7(2019), No. 4, 500-509
https://doi.org/10.22364/bjmc.2019.7.4.04

Subword Segmentation for Machine Translation Based on Grouping Words by Potential Roots

Jānis ZUTERS, Gus STRAZDS

University of Latvia, Raina blvd. 19, LV-1586 Riga, Latvia

janis.zuters@lu.lv, gstrazds@gmail.com

Abstract. This paper proposes a new subword segmentation method for machine translation. The algorithm, which we call GenSeg, is generic in the sense that it can be applied to any language, but is designed with an emphasis on inflectional splitting, i.e. it attempts to split words on boundaries corresponding to inflectional suffixes. The main principle of the method is grouping together words that share a common middle substring, and then separating the best such substring from the rest of the word. GenSeg is a cross-language method extended with some language-specific morphological analysis rules (currently for the Latvian language). To verify its effectiveness, we performed machine translation experiments in two directions: Latvian-English and English-Latvian, obtaining minor improvements in translation quality when using our pre-processing method.

Keywords: neural machine translation, word segmentation, morphological analysis

1. Introduction

Data sparsity remains one of the biggest challenges in neural machine translation, especially for morphologically rich languages with relatively small available parallel training data (Pinnis et al., 2017a). One of most common techniques to address this problem is that text preprocessing for subword segmentation. This article proposes the GenSeg word splitting method, which is a fully automatic method to perform morphologically motivated machine-translation-oriented word splitting. This work follows up on our previous research (Zuters et al., 2018), here additionally applying some explicit morphological analysis to try to obtain better word splittings.

By construction and purpose, available segmentation algorithms vary from “pure technical” and machine-translation-focused, such as BPE (Sennrich et al., 2016), to morphology-motivated, like Lemming (Müller et al., 2015). The proposed GenSeg algorithm is implicitly based on the statistics of occurrence of common parts of words in word forms found in the text, on the assumption that more frequent parts are more likely to be word roots, and thus should be split out for subword segmentation.

The output text from the GenSeg algorithm partly resembles morphologically segmented text (especially in Latvian-specific mode), but we make no claims to the splits being linguistically meaningful. On the contrary, for better machine translation results, the algorithm incorporates design decisions that intentionally deviate from more linguistically correct splittings. We assess the performance of our proposed algorithm by comparing end-result translation quality (as measured by BLEU score) when input text is preprocessed using GenSeg vs. preprocessing using only BPE segmentation. We chose BPE as a baseline for comparison because BPE, since its introduction a few years ago, has become very widely supported and used, and currently seems to be pretty much the default subword segmentation method for machine translation.

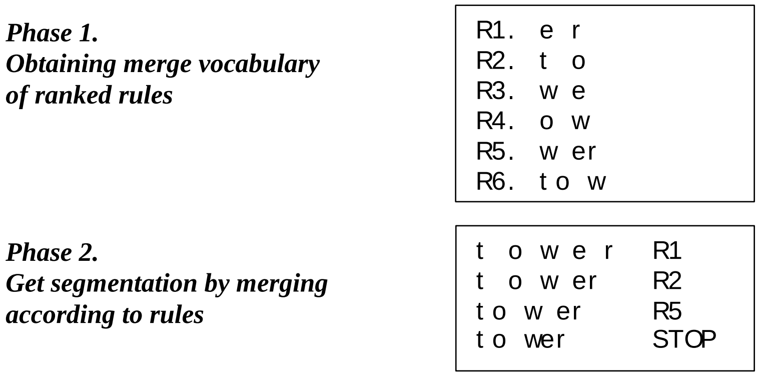
2. Related work

This section describes several subword segmentation algorithms which are (at least partially) designed for preprocessing input text for neural machine translation.

2.1. BPE

The Byte Pair Encoding (BPE) algorithm for subword segmentation, proposed by Sennrich et al. (2016), is among the simplest and thus fastest and most generic text segmentation algorithms. It is completely generic / language agnostic and makes no claims about morphological adequacy. It uses an iterative process to calculate statistics of the most frequent character sequences and a ranked sequence of candidate merge pairs, which is then used to segment words in a similar way – starting with the input text represented as a sequence of individual characters, the most frequent merge pairs are greedily merged to form “supercharacters” or subword tokens, until the desired vocabulary size is reached (see example in Fig. 1).

The design of the algorithm provides for control over the effective size of the token vocabulary of a processed text -- a valuable feature since the trade-off between maximum input sequence length vs total vocabulary size is an important hyperparameter for neural machine translation systems.



Phase 1.
Obtaining merge vocabulary of ranked rules

```
R1. e r
R2. t o
R3. w e
R4. o w
R5. w er
R6. to w
```

Phase 2.
Get segmentation by merging according to rules

```
t o w e r   R1
t o w er    R2
to w er     R5
to wer      STOP
```

Fig. 1. An illustrative example of segmentation of word “tower” with BPE.

BPE has become a benchmark subword segmentation algorithm for pre-processing text for machine translation, and, due to the control it provides over vocabulary size, is also sometimes used as a supplemental pre-processing utility after some other subword segmentation tool has been run (we also use it in this manner to postprocess the output from our GenSeg algorithm).

2.2. Morfessor

Morfessor (Virpioja et al., 2013) is tool for probabilistic machine-learning-based morphological segmentation, sometimes also used for machine translation.

The main principle of training in Morfessor is a recursive process of examining all possible two-part segmentations of word units until a special cost stops decreasing. Once the model is obtained, an extension of the Viterbi algorithm (Viterbi, 1967) is used to find the best segmentation for each word.

2.3. PRPE

PRPE (Prefix-Root-Postfix-Encoding) subword segmentation (Zuters et al., 2018) exploits the 'root alignment' principle to extract ranked lists of potential prefixes, roots, and postfixes from a text corpus, which are later combined to obtain word segmentations.

The main idea behind the algorithm for collecting building blocks for segmentation is that left substrings of words are assumed to be potential roots, and matching these potential roots with the middle substrings of other words (see Fig. 2) yields information for extracting also potential prefixes and postfixes.

The algorithm is almost language-independent, requiring only a small amount of language-specific code to adapt it to a particular language.

<table>
<tr><th>u</th><th>n</th><th>b</th><th>e</th><th>l</th><th>i</th><th>e</th><th>v</th><th>a</th><th>b</th><th>l</th><th>e</th><th>s</th></tr>
<tr><td colspan="2">prefix</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="4">potential root</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="5">potential root</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="6">potential root</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="7">potential root</td><td></td><td></td><td></td><td></td></tr>
</table>

Fig. 2. The illustration of the 'Root alignment' principle in word "unbelievables": potential roots aligned with the middle part of the word to collect statistics for prefix "un" (Zuters et al., 2018).

2.4. Lemming

Lemming (Müller et al., 2015) is a data driven word lemmatization algorithm based on Chrupala (2006) which treats lemmatization as a classification task.

The algorithm follows (Chrupala, 2006) in computing shortest edit scripts for converting words to their lemmatized forms. In addition, they use training data annotated with a set of textual features hand-crafted for this task plus part-of-speech tags to train a machine learning model to jointly predict both lemmatized forms and a morphological analysis. This model can then be applied to classify previously unseen words to obtain the appropriate edit script for converting the word to its lemmatized form. It is reported that the algorithms works better for suffixal morphology (where most of the variation in related word forms occurs in the endings of words).

3. GenSeg Segmentation Method

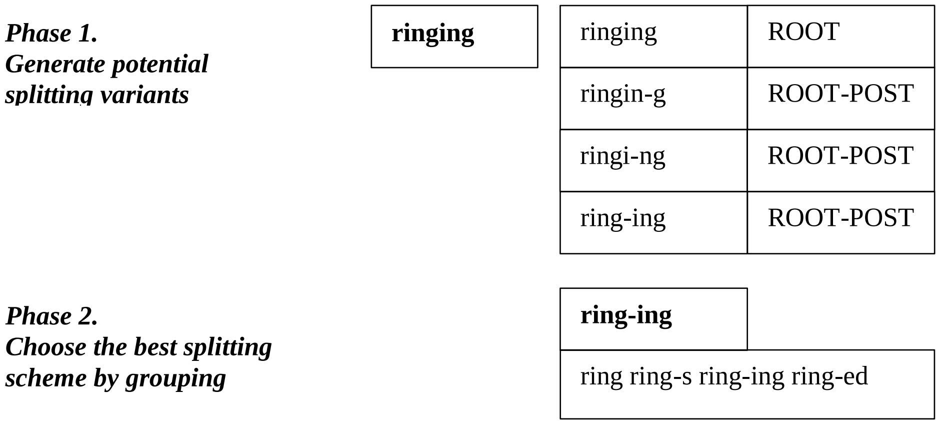
This section describes the proposed GenSeg method[1] for morphologically motivated subword segmentation. The proposed solution is aimed at applications in machine translation and comes in two modes – Latvian language specific and generic.

3.1. General Overview

The main principle of the method is grouping words by potential roots (or to be more general – by potential technical word lexical forms) thus finding the most valuable potential root for each word and by this also the pattern of splitting it. The value of a potential root depends on how many different word forms found in the training corpus can be associated with it.

For simplicity and also due to the specific requirements of the task of machine translation (where too much segmentation decreases translation quality because of limitations in the ability of currently used architectures to handle input and output sequences that are too long), as well as the specifics of construction of the method, GenSeg comes with two restrictions:

- Splitting of words is performed into at most three parts: prefix – root (middle part) – postfix (ending), e.g., compounds are not split into separate roots (see segmentation example in Table 1).
- A word is left unsplit if there are not enough word forms related to it in the corpus.



Phase 1.
Generate potential splitting variants

ringing

ringing	ROOT
ringin-g	ROOT-POST
ringi-ng	ROOT-POST
ring-ing	ROOT-POST

Phase 2.
Choose the best splitting scheme by grouping

ring-ing

ring ring-s ring-ing ring-ed

Fig. 3. An illustrative example of segmentation process of a word with GenSeg.

[1] Source code available at: `https://github.com/zuters/genseg`

Segmentation with GenSeg is carried out in two phases (see example in Fig. 3):

- Obtain all potential splitting schemes and (technical) lexical forms for each word (a simplified form of lemmatization).
- Group together words of the same potential lexical form to find for each word the most plausible base form, and choose the splitting scheme accordingly (the best splitting scheme for each word is voted as related to the most representative lexical form within the text corpus).

Table 1. Segmentation examples with GenSeg.

Latvian-specific GenSeg on Latvian text	At– raktīv –ajam jamaikiešu iz– celsm –es tenor –am Džermein –am Smit –am , ko Andr –is Pog –a savulaik no– lūk –ojis Boston –as ie– stud –ējumā , narkodīler –a Sportinlaifa tēl –s ir firm –as lom –a četr –os kontinent –os .
Generic GenSeg on English text	Man –y of the student –s wer –e young people in rural are –as who receiv –ed tuition online , but also had to make frequent visi –ts to the colle –ge in Inver –ness from where the –y live –d in place –s such as Lochaber .

3.2. Obtaining Potential Splitting Variants

In this phase, for each word, all potential splitting schemes are generated, while also tagging each split unit as prefix, root, or postfix (see example in Fig. 3). As our previous experiments with segmentation for machine translation showed that translation quality suffers with too much splitting – an effect which can outweigh potential benefits from a full, correct morphological splitting, we only split words into three parts at most: optional prefix, mandatory root (main part), optional postfix. What's more, empirically we subsequently got our best results by splitting only the postfix from the rest of the word.

We have implemented two slightly different algorithms for collecting splitting schemes – generic and Latvian specific.

The generic splitting collection algorithm exploits the main idea of the PRPE segmenter (Zuters et al., 2018) – using the ‘Root alignment’ principle (see Section 2.3.) with simplifications in two aspects:

- Restriction of the splitting rate to three units,
- No language specific source code.

The splitting collection algorithm for Latvian makes use of the following language specific components:

- A rule set for postfix recognition;
- A list of possible prefixes;
- A root generation scheme to facilitate recognition of words with different roots as belonging to the same lexical form.

Although the latter component does not directly impact the splitting itself, it provides the system with additional information about relations between word forms and lexical forms, useful in the second phase of the method.

3.3. Grouping Words by Potential Roots

In this phase, words are grouped by potential lexical forms (in the simplest case – roots) to create a rating for each lexical form. Then, the most plausible splitting results from the most highly rated lexical form related to the word determine, which splitting scheme is chosen.

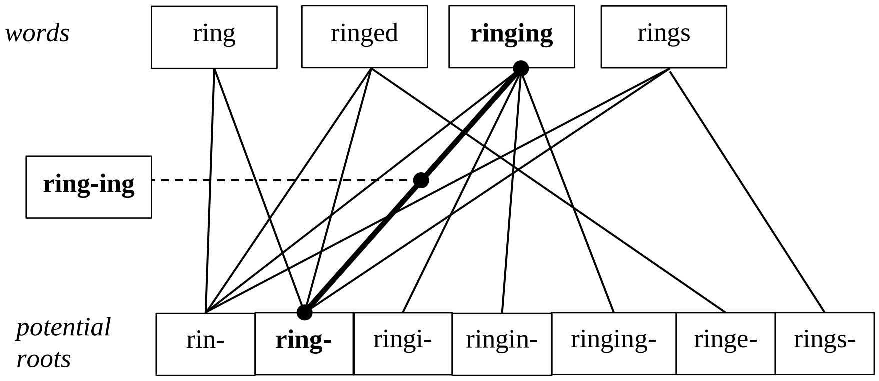


Fig. 4. A simplified example of word grouping by potential roots.

The conceptual algorithm of word grouping consists of the following steps (for illustration, see Fig. 4):

1. From phase 1 (Section 3.2), a relation between words and potential lexical forms (roots) was obtained.
2. A potential root is rated by the sum of the respective word-root rates (represented as lines in the illustration) – the more word forms the root is associated with the bigger is its potential value.
   a. The word-root rate for the generic method depends on additional information obtained in the phase 1:
      i. root rank – how plausible the root is (more common roots are preferred),
      ii. postfix rank – how plausible is the postfix (more common postfixes are preferred),
      iii. root length (longer roots are preferred)
   b. The word-root rate for the Latvian-specific method depends on additional information about the function of the corresponding root in the word (also obtained during phase 1).
3. The most valuable root (lexical form) and, accordingly, the split scheme, is associated to each word. (Note that no syntactic information is used in this process).

From the simplified example with word ‘ringing’ (Fig. 4) we see that the most “popular” root candidates are “rin-” and “ring-”, but the latter (together with the corresponding splitting scheme “ring-ing”) is preferred because the potential root is longer.

4. Experiments and Results

The main objective of our experiments was to test whether pre-processing corpora with GenSeg yields better machine translation results relative to baseline segmentation using just BPE (Sennrich et al., 2016). BLEU score (Papineni et al., 2002) was used to evaluate results.

For our experiments, we used the English-Latvian dataset from the WMT 2017[2] shared task in news translation, and trained models for both translation directions: English-to-Latvian (en-lv) and Latvian-to-English (lv-en). In our previous experiments with the PRPE segmenter (Zuters et al., 2018), we observed comparatively less improvement of translation quality in the en-lv direction – i.e., towards the more inflected language.

The approximate size of each of the parallel corpora is 1.6M sentences. As a starting point, we use the data as pre-processed (filtered, normalised, tokenised) by Pinnis et al. (2017b) for their experiments.

We produced subword segmented versions of both the English and Latvian texts using several configurations of GenSeg (Latvian text segmented with Latvian specific version, English text segmented with generic version with minimum allowed root length = 3). As a baseline, we processed the same corpora using just BPE[3]. All the GenSeg segmentations were also post-preprocessed using BPE, to obtain, for all cases, corpora with approximately comparable token vocabulary sizes (counting the number of unique subword tokens occurring in the training datasets). Note that this post-pre-processing step (applying BPE segmentation to the GenSeg segmented data) results in further sub-segmentation of the root, postfix, and prefix pieces of words produced by GenSeg, but does not merge subword pieces across the boundaries introduced by GenSeg.

After a machine translation model is trained and used to generate output sequences (of subword tokens) corresponding to the input sequences, these pre-processing steps are undone in reverse order: first BPE-segmented subword tokens are merged, then GenSeg splits are merged, to finally obtain word sequences for the translations.

In the initial phase of the experimentation with Latvian segmentation, we tried two variants:

- Segmentation into three parts (prefix-root-postfix);
- Segmentation into two parts (root-postfix), without splitting away prefixes.

Although human review of the segmented text of the three-part variant looked promising, the translation quality (as measured by BLEU score) using this segmentation scheme decreased, so we excluded three-part segmentation from our further experiments.

For final experimentation, to obtain the results reported below, we used the following segmented texts in English and Latvian:

- BPE(lv) – BPE segmentation on Latvian text (joint BPE vocab en+lv);
- BPE(en) – BPE segmentation on English text (joint BPE vocab en+lv);
- lvseg(lv) – Latvian-specific GenSeg on Latvian text (+BPE);
- genseg(en) – generic GenSeg on English text (+BPE);

[2] *http://www.statmt.org/wmt17/translation-task.html*

[3] *https://github.com/rsennrich/subword-nmt*

To evaluate the impact of GenSeg on machine translation, we then used these variously segmented parallel corpora to train English-to-Latvian (en-lv) and Latvian-to-English (lv-en) translation models using the FairSeq[4] NMT system with Transformer models (Vaswani et al. 2017)[5].

The translation results are given in Tables 2 and 3.

Table 2. Translation results with different segmentation techniques (en-lv).

Experiment ID	Text From	Text To	BLEU	p-val vs BPE
BPE(en-lv)	BPE(en)	BPE(lv)	23.52	
GenSegLV(en-lv)	BPE(en)	lvseg(lv)	23.74	0.14
GenSegBoth(en-lv)	genseg(en)	lvseg(lv)	23.72	0.15

Table 3. Translation results with different segmentation techniques (lv-en).

Experiment ID	Text From	Text To	BLEU	p-val vs BPE
BPE(lv-en)	BPE(lv)	BPE(en)	25.53	
GenSegLV(lv-en)	lvseg(lv)	BPE(en)	25.72	0.16
GenSegBoth(lv-en)	lvseg(lv)	genseg(en)	25.82	0.04

Previously published results (Pinnis et al., 2017b; Zuters et al., 2018) have shown that the translation direction English-to-Latvian in general yields worse scores than Latvian-to-English, and in all cases our results confirm this finding. This could be explained both by the assumption that translation towards a morphologically richer language is a more challenging task as such (because the translation model needs to generate appropriately inflected word forms), and by the observation that word order can be less strict in more highly inflected languages (but reordered word sequences can negatively impact BLEU scores even when they may in fact be acceptable in the target language).

[4] *https://github.com/facebookresearch/fairseq-py*

[5] We used one of the standard transformer configurations included in the Fairseq library ('transformer_wmt_en_de'), and trained our models on a single computer with 4 Tesla V100 GPUs (using the command line argument --update-freq=2 to accumulate gradients across 2 minibatches to simulate training with 8 GPUs). Output translations were generated using a beam search width of 12, from a model obtained by running *scripts/average_checkpoints.py* to merge 5 epoch checkpoints – two preceding and two following the checkpoint that achieved the best validation score during training.

Our experiments with GenSeg demonstrated minor improvements in machine translation metrics, but only for the output of GenSegBoth(lv-en) (Table 2) did we obtained a statistically significant improvement[6].

5. Conclusion

In this paper, we propose an algorithm for automatic word splitting as a preprocessing step for machine translation. The experimental results show the GenSeg algorithm contributing small improvements to machine translation quality. Results also show that machine translation of morphologically rich languages still remains challenging compared to analytic languages.

Additionally, our results affirm that splitting into fewer parts (prior to subword segmentation with BPE) gives better results, even when splitting into more parts might be morphologically adequate. In practice this means that the best results are achieved if words are split into no more than two parts (splitting word postfixes away from a root with prefixes).

Obtained improvements in translation quality with GenSeg pre-splitting were not large, in several cases falling below a commonly used threshold for statistical significance. These results together with our previous ones could be interpreted as an indication that cardinally different text pre-processing approaches might be required to achieve more improvements.

Acknowledgements

The research has been supported by the European Regional Development Fund within the research project “Neural Network Modelling for Inflected Natural Languages” No. 1.1.1.1/16/A/215, and the Faculty of Computing, University of Latvia.

References

Chrupala, G. (2006). Simple data-driven context-sensitive lemmatization. *Procesamiento del Lenguaje Natural*, (37):121–127.

Müller, T, Cotterell, R. Fraser, A. Schütze, H. (2015). Joint Lemmatization and Morphological Tagging with Lemming, In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, pp. 2268-2274. Lisbon, Portugal.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J. (2002). BLEU: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp. 311–318.

Pinnis, M., Krišlauks, R., Deksne, D., Miks, T. (2017a). Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In: *Ekštein K., Matoušek V. (eds) Text, Speech, and Dialogue. TSD 2017. Lecture Notes in Computer Science*, vol. 10415. Springer, Cham.

[6] Statistical significance was estimated via bootstrap resampling using the script *analysis/bootstrap-hypothesis-difference-significance.pl* from the Moses MT system: `https://github.com/moses-smt/mosesdecoder`

Pinnis, M., Krišlauks, R., Miks, T., Deksne, D., Šics, V. (2017b). Tilde's Machine Translation Sys-tems for WMT 2017. In: *Proceedings of the Second Conference on Machine Translation (WMT 2017)*, Volume 2: Shared Task Papers (pp. 374–381). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W17-4737.

Sennrich, R., Haddow, B., Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL2016)*. Berlin, Germany.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention Is All You Need. In: *Advances in Neural Information Processing Systems (NIPS 2017)*.

Virpioja S., Smit P., Grönroos, S.-A., Kurimo, M. (2013). Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline. In: *Aalto University publication series SCIENCE + TECHNOLOGY*, 25/2013. Aalto University.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*. 13 (2), pp. 260–269.

Zuters, J., Strazds, G., Immers, K. (2018). Semi-Automatic Quasi-Morphological Word Segmentation for Neural Machine Translation. *In proceedings of the 13th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2018)*, pp. 289-301. Trakai, Lithuania.

Received August 12, 2019, revised October 30, 2019, accepted December 20, 2019