# Morphological Segmentation Can Improve Syllabification

**Garrett Nicolai Lei Yao Grzegorz Kondrak**
Department of Computing Science
University of Alberta
{nicolai,lyao1,gkondrak}@ualberta.ca

## Abstract

Syllabification is sometimes influenced by morphological boundaries. We show that incorporating morphological information can improve the accuracy of orthographic syllabification in English and German. Surprisingly, unsupervised segmenters, such as Morfessor, can be more useful for this purpose than the supervised ones.

## 1 Introduction

Syllabification is the process of dividing a word into syllables. Although in the strict linguistic sense syllables are composed of phonemes rather than letters, due to practical considerations we focus here on orthographic syllabification, which is also referred to as *hyphenation*. Some dictionaries include hyphenation information to indicate where words may be broken for end-of-line divisions, and to assist the reader in recovering the correct pronunciation. In many languages the orthographic and phonological representations of a word are closely related.

Orthographic syllabification has a number of computational applications. Incorporation of the syllable boundaries between letters benefits grapheme-to-phoneme conversion (Damper et al., 2005), and respelling generation (Hauer and Kondrak, 2013). Hyphenation of out-of-dictionary words is also important in text processing (Trogkanis and Elkan, 2010). Because of the productive nature of language, a dictionary look-up process for syllabification is inadequate. Rule-based systems are generally outperformed on out-of-dictionary words by data-driven methods, such as those of Daelemans et al. (1997), Demberg (2006), Marchand and Damper (2007), and Trogkanis and Elkan (2010).

Morphological segmentation is the task of dividing words into morphemes, the smallest meaning-bearing units in the word (Goldsmith, 2001). For example the morpheme *over* occurs in words like *hold+over*, *lay+over*, and *skip+over*.[1] Roots combine with derivational (e.g. *refut+able*) and inflectional affixes (e.g. *hold+ing*). Computational segmentation approaches can be divided into rule-based (Porter, 1980), supervised (Ruokolainen et al., 2013), semi-supervised (Grönroos et al., 2014), and unsupervised (Creutz and Lagus, 2002). Bartlett et al. (2008) observe that some of the errors made by their otherwise highly-accurate system, such as *hol-dov-er* and *coad-ju-tors*, can be attributed to the lack of awareness of morphological boundaries, which influence syllabification.

In this paper, we demonstrate that the accuracy of orthographic syllabification can be improved by considering morphology. We augment the syllabification approach of Bartlett et al. (2008), with features encoding morphological segmentation of words. We investigate the degree of overlap between the morphological and syllable boundaries. The results of our experiments on English and German show that the incorporation of expert-annotated (*gold*) morphological boundaries extracted from lexical databases substantially reduces the syllabification error rate, particularly in low-resource settings. We find that the accuracy gains tend to be preserved when unsupervised segmentation is used instead. On the other hand, relying on a fully-supervised system appears to be much less robust, even though it generates more accurate morphological segmentations than the unsupervised systems. We propose an explanation for this surprising result.

[1] We denote syllable boundaries with '-', and morpheme boundaries with '+'.

*Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 99–103,
Berlin, Germany, August 11, 2016. ©2016 Association for Computational Linguistics

## 2 Methods

In this section, we describe the original syllabification method of Bartlett et al. (2008), which serves as our baseline system, and discuss various approaches to incorporating morphological information.

### 2.1 Base system

Bartlett et al. (2008) present a discriminative approach to automatic syllabification. They formulate syllabification as a tagging problem, and learn a Structured SVM tagger from labeled data (Tsochantaridis et al., 2005). Under the Markov assumption that each tag is dependent on its previous $n$ tags, the tagger predicts the optimal tag sequence (Altun et al., 2003). A large-margin training objective is applied to learn a weight vector to separate the correct tag sequence from other possible sequences for each training instance. The test instances are tagged using the Viterbi decoding algorithm on the basis of the weighted features.

Each training instance is represented as a sequence of feature vectors, with the tags following the "Numbered NB" tagging scheme, which was found to produce the best results. In the scheme, the B tags signal that a boundary occurs after the current character, while the N tags indicate the distance from the previous boundary. For example, the word *syl-lab-i-fy* is annotated as: N1 N2 B N1 N2 B B N1 N2. The feature vectors consist of all $n$-grams around the current focus character, up to size 5. These $n$-grams are composed of context letters, and word-boundary markers that are added at the beginning and end of each word.

### 2.2 Morphological information

We incorporate available morphological information by adding morpheme boundary markers into the input words. The extracted features belong to two categories: orthographic and morphological. The orthographic features are identical to the ones described in Section 2.1. The morphological features are also contextual $n$-grams, but may contain morphological breaks, which can potentially help identify the correct syllabification of words. Manually-annotated morphological lexicons sometimes distinguish between inflectional, derivational, and compound boundaries. We can pass this information to the syllabification system by marking the respective boundaries with different symbols.

Since morphologically annotated lexicons are expensive to create, and available only for well-studied languages, we investigate the idea of replacing them with annotations generated by fully-supervised, distantly-supervised, and unsupervised segmentation algorithms.

#### 2.2.1 Fully-supervised

While supervised methods typically require large amounts of annotated training data, they can perform segmentation of unseen (out-of-dictionary) words. As our fully-supervised segmenter, we use the discriminative string transducer of Jiampojamarn et al. (2010). The transducer is trained on aligned source-target pairs, one pair per word; the target is identical to the source except that it includes characters that represent morphological breaks. Using source and target context, the transducer learns to insert these breaks into words.

#### 2.2.2 Distantly-supervised

Whereas morphologically-annotated lexicons are rare, websites such as Wiktionary contain crowd-generated inflection tables for many languages. A distantly-supervised segmenter can be trained on semi-structured inflection tables to divide words into stems and affixes without explicit segmentation annotation. We adopt the approach of Nicolai and Kondrak (2016), which combines unsupervised alignment with a discriminative string transduction algorithm, An important limitation of this approach is that it can only identify inflectional morpheme boundaries.

#### 2.2.3 Unsupervised

Unsupervised methods have the advantage of requiring no training data. We investigate the applicability of two unsupervised segmenters: Morfessor (Creutz and Lagus, 2005) and Morpheme++ (Dasgupta and Ng, 2007). Morfessor uses the minimum description length (MDL) principle to predict a word as a likely sequence of morphemes. Since the baseline version of Morfessor tends to over-segment rare words, we instead apply Morfessor FlatCat (Grönroos et al., 2014), which reduces over-segmentation through the use of a hidden Markov model. Morpheme++ is another system that is capable of distinguishing between prefixes, suffixes, and stems by taking advantage of the regularity of affixes.

## 3 Experiments

In this section, we introduce our data sets, and discuss the overlap between morphological and syllabic boundaries. We investigate the quality of the morphological segmentations of produced by various methods, and replicate the syllabification results of Bartlett et al. (2008). Finally, we discuss the results of incorporating morphological information into the syllabification system.

### 3.1 Data

Our data comes from the English and German sections of the CELEX lexical database (Baayen et al., 1995). The English and German training sets contain 43,212 and 41,382 instances, with corresponding development sets of 8,735 and 5,173 instances, and test sets of 8,608 and 5,173 instances. The distantly-supervised and fully-supervised segmenters were trained on the union of the training and development sets, while the unsupervised segmenters were applied to the union of the training, development and test sets. The distantly-supervised system had no access to the gold morphological segmentations.

The annotation in CELEX distinguishes between inflectional vs. derivational affixes, as well as derivational vs. compound breaks. The latter distinction did not help in our development experiments, so we disregard it. We refer to the two subsets of the morpheme boundary annotations as "Gold Inflectional" and "Gold Derivational".

### 3.2 Quality of morphological segmentation

Table 1 shows the word accuracy (entire words segmented correctly) of various segmentation methods on the test sets. Unsurprisingly, the fully-supervised segmenter is substantially more accurate than the other systems. The distantly-supervised system can only identify inflectional boundaries. so its overall accuracy is rather low; however, its accuracy on the inflectional boundaries is 96.0% for English, and 82.6% for German. Among the unsupervised systems, Morfessor FlatCat is only slightly better than Morfessor 1.0, while Morpheme++ is comparable on German, and significantly better on English. It should be noted that since our focus is on syllabification, no careful parameter tuning was performed, and our data excludes word frequency information.

| | EN | DE |
|---|---|---|
| Morfessor 1.0 | 59.4 | 39.8 |
| Morfessor FlatCat | 59.6 | 40.8 |
| Morpheme++ | 66.3 | 39.1 |
| Distantly-supervised | 63.5 | 21.3 |
| Fully-supervised | 95.4 | 71.3 |

Table 1: Morphological segmentation word accuracy on the test set.

| | EN | DE |
|---|---|---|
| Morfessor | 38.2 | 61.4 |
| Morfessor FlatCat | 39.1 | 66.7 |
| Morpheme++ | 46.4 | 67.1 |
| Distantly-supervised | 24.8 | 7.9 |
| Fully-supervised | 44.5 | 51.5 |
| Gold | 45.1 | 49.7 |
| Gold Inflectional | 24.4 | 4.5 |
| Gold Derivational | 68.6 | 57.6 |

Table 2: Overlap between syllabic and morphological boundaries on the test set.

Table 2 shows the percentage of the predicted morphological breaks that match gold syllable boundaries. We observe that the inflectional boundaries are far less likely than the derivational ones to correspond to syllable breaks. We also note that on German the unsupervised segmenters exhibit much higher syllabification overlap than the gold annotation. We attribute this to the tendency of the unsupervised methods to oversegment.

### 3.3 Baseline syllabification

As a baseline, we replicate the experiments of Bartlett et al. (2008), and extend them to low-resource settings. Since the training sets are of slightly different sizes, we label each training size point as specified in Table 3. We see that correct syllabification of approximately half of the words is achieved with as few as 100 English and 50 German training examples.

### 3.4 Morphologically-informed syllabification

Our main set of experiments concerns the incorporation of the morphological information obtained from methods described in Section 2.2 into the baseline syllabification system. As seen in Table 3, the accuracy of the baseline syllabification system trained on a large number of instances is already very high, so the gains introduced by mor-

| Label | Training Size | | Error Rate | |
|---|---|---|---|---|
| | EN | DE | EN | DE |
| A | 51 | 45 | 61.27 | 52.97 |
| B | 101 | 91 | 51.25 | 44.08 |
| C | 203 | 182 | 43.05 | 35.37 |
| D | 406 | 364 | 34.00 | 25.32 |
| E | 812 | 727 | 27.23 | 19.01 |
| F | 1623 | 1455 | 21.50 | 12.74 |
| G | 3247 | 2910 | 16.96 | 9.24 |
| H | 6493 | 5819 | 10.50 | 6.27 |
| I | 12987 | 11639 | 6.61 | 4.64 |
| J | 25974 | 23278 | 3.73 | 3.19 |
| K | 51947 | 46555 | 2.18 | 2.04 |

Table 3: Absolute error rate for the baseline with varying amounts of the training data.

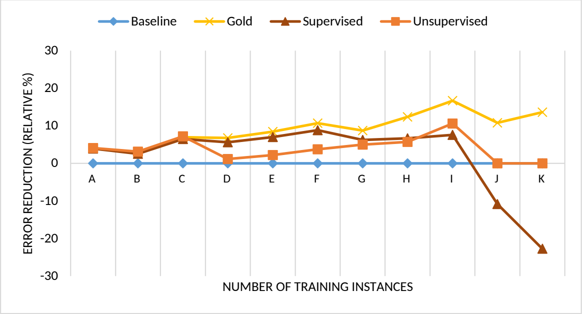

Figure 1: Syllabification error rate reduction on English.

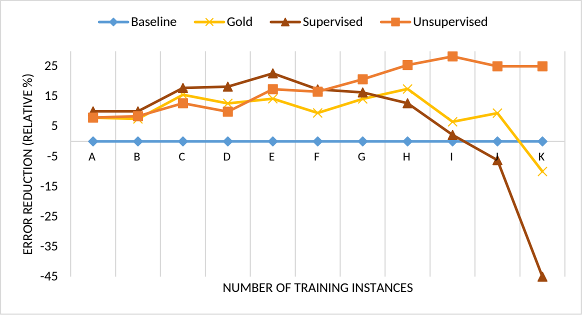

Figure 2: Syllabification error rate reduction on German.

phology are necessarily small. In Figures 1 and 2, we show the relative error reduction at various training sizes. The absolute error rate can be obtained by multiplying the values from the table and the figures.

For the sake of clarity, we omit some of the methods from the graphs. The unsupervised methods are represented by Morfessor FlatCat. The distantly-supervised system is generally successful at predicting the inflectional boundaries, but fails to improve on the baseline, as they are less important for syllabification than the derivational boundaries.

### 3.5 Discussion

Overall, the results confirm that morphology can help syllabification. The incorporation of gold segmentation boundaries consistently leads to the reduction of the syllabification error rate; the only exception occurs on the full English training set. While the fully-supervised system provides a benefit at lower training thresholds, it actually hurts the accuracy at larger training sizes. Notably, unsupervised segmentation appears to outperform fully-supervised segmentation as the amount of the training data increases; the corresponding error rate reduction approaches 25% on German.

One explanation for the strong performance of the unsupervised systems is their high accuracy on compound words. Consider the German compound *Toppflagge* “masthead flag”. An unsupervised system is able to guess that the word is composed of the words *Topp* and *Flagge* that exist in the lexicon on their own. To produce the same segmentation, the fully-supervised system must be trained on a number of compound words that include either *topp* or *flagge*. Since compound boundaries are almost always syllable breaks as well, they have a strong effect on syllabification.

Sometimes even a linguistically incorrect segmentation proposed by an unsupervised segmenter may work better for the purposes of syllabification. Many words of Latin origin contain affixes that are no longer productive in English. Thus, an unsupervised system over-segments the word *ob+literate*, which allows it to produce the correct syllabification *ob-lit-er-ate*, as opposed to *o-blit-er-ate* predicted by the gold-informed system. This phenomenon appears to be particularly frequent in German.

## 4 Conclusion

We have demonstrated that morphological information can improve the accuracy of orthographic syllabification. We have found that unsupervised segmentation methods often perform better than supervised methods, and can rival gold human annotation. We have proposed two explanations for this counter-intuitive phenomenon. We hope that this work will contribute a computational perspective on the issue of interaction between syllabification and morphology.

## Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the Alberta Innovates Technology Futures.

## References

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov support vector machines. In *ICML*, pages 3–10.

Harald R. Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania.

Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. 2008. Automatic syllabification with structured SVMs for letter-to-phoneme conversion. In *ACL*, pages 568–576.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*, pages 51–59.

Walter Daelemans, Antal van den Bosch, and Ton Weijters. 1997. Igtree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intellegence Review*, 11(1-5):407–423.

Robert I Damper, Yannick Marchand, J-DS Marsters, and Alexander I Bazin. 2005. Aligning text and phonemes for speech technology applications using an em-like algorithm. *International Journal of Speech Technology*, 8(2):147–160.

Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *HLT-NAACL*, pages 155–163.

Vera Demberg. 2006. Letter-to-phoneme conversion for a german text-to-speech system.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguisitics*, 27(2), June.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *COLING*, pages 1177–1185.

Bradley Hauer and Grzegorz Kondrak. 2013. Automatic generation of English respellings. In *NAACL*, pages 634–643.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training network. In *NAACL*.

Yannick Marchand and Robert I. Damper. 2007. Can syllabification improve pronunciation by analogy of English? *Natural Language Engineering*, 13(1):1–24.

Garrett Nicolai and Grzegorz Kondrak. 2016. Leveraging inflection tables for stemming and lemmatization. In *ACL*.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *CoNLL*.

Nikolaos Trogkanis and Charles Elkan. 2010. Conditional random fields for word hyphenation. In *ACL*, pages 366–374.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484.