JB132 submission to the SIGMORPHON 2022 Shared Task 3 on Morphological Segmentation

Jan Bodnár
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic
bodnar@ufal.mff.cuni.cz

Abstract

This paper describes the JB132 submission to the SIGMORPHON 2022 Shared Task 3 on Morpheme Segmentation. In this paper we describe probabilistic model trained with the Expectation-Maximization algorithm, we provide the results and analyze sources of errors and general limitations of our approach. The model was implemented within our own modular probabilistic framework.

1 Introduction

This paper describes JB132 submission to Shared Task on Morphological segmentation, which is the task of segmentation of words to the smallest units carrying meaning - morphemes (e.g. prefixes, root, suffixes).

Our general approach was to create our own modular framework for probabilistic models trained via Expectation-Maximization, so that we can quickly test large number of various model architectures.

We designed various probabilistic models, described them within the framework and tested them across languages. In this paper we provide the description of the best model architecture. Since the algorithm achieves poor results, we further analyze its outputs and describe causes of errors it makes.

2 Task

The Shared Task focused on both morphological segmentation of solitary words (Task 1) and words in sentences (Task 2), but we have only participated in Task 1. The training data spanned across 9 languages (Czech, English, French, Hungarian, Spanish, Italian, Latin, Russian, Mongolian) and contained tens of thousands to hundreds of thousands training samples.

The structure and complexity of input data varied. The Czech words were segmented to morphs (*absorbovat ab-sorb-ova-t*), while e.g. Spanish and Russian data contained segmentation to morphemes, including change of root and presence of morphemes that were used in the derivation of the word but now only map to null morphs (encuestéis encuesta-ar-éis; автоматизируемые автоматизм-ировать-уем-ый-ые).

3 Related Work

Probabilistic models are commonly used in morphological segmentation, although often focused on morphs instead of morphemes and trained in unsupervised or weakly supervised settings. There are three (sometimes overlapping) groups of probabilistic models used for segmentation: the first group are Bayesian models, which rely on complex generative stories, including even prior distributions of numbers of morphemes of words or prior distribution of morpheme frequencies. An interesting example of this approach is (Snyder and Barzilay, 2008), which experimented with a joint multilingual model for several related languages and showed that it can improve the resulting segmentation in unsupervised setting.

The second group are Maximum a posteriori probability (= Minimum description length) models. These models try to find the best compression of words (including the size of the compression model's parameters) and are usually optimized via some kind of local optimization. Models of this type are e.g. (Creutz and Lagus, 2002) and (Goldsmith, 2006), which also use morpheme lexicons, but unlike our model consider size of the dictionary part of the loss function.

The last group are Expectation-Maximization models such as (Creutz and Lagus, 2004),(Grönroos et al., 2020), which tend to make use of simpler loss functions that further simplify when EM is applied and thus allow for faster computation.

19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 152 - 156

4 Solution

Our approach was to create a modular framework for probabilistic models optimized via the Expectation-Maximization algorithm. We then described various architectures within this framework and tried to find the one that works the best across the languages.

Our final architecture consists of two main parts: word→morpheme generator (Fig. 1) which models the structure of words as sequence of morphemes and the morpheme→morphs model (Fig. 2) which models the morpheme realization.

Each of those parts is trained separately and they are then merged together.

4.1 Word→Morphemes model

The goal of this model is to learn the high level structure of the word. The final version of this model (Fig. 1) uses Hidden Markov Model with hidden three states - prefix, root, suffix (only the transitions to the same or later state are allowed). Each of the states has an independent output model - Prefix outputs either one of the prefix morphemes in its morpheme dictionary or a string generated by a letter unigram character model (to generate the unknown morphemes). Root and suffix work on the same principle.

The morpheme dictionaries were obtained from the training dataset using a simple heuristic for identification of the root morpheme (roots tend to be long and infrequent compared to the affixes. Morphemes in front of a root are prefixes, morphemes behind it are suffixes).

After initialization, the model was trained on the second column of the dataset - we simply concatenated the morphemes and trained the model to split them back. This allowed the model to learn the morphemic structure of words.

The model is trained via EM - we first let the model find the most probable way of generating the word in a recursive manner: If we ask some module to generate subword beginning with i-th letter, then it uses itself and its submodules to find the most likely ways of generating the following 1, 2, 3, ... letters. Then it returns us the descriptions of such ways of generation.

With this recursive principle the top-most module will give us the likelihood of the best generation of the whole word and the recursive description (tree) describing how the modules generated it (e.g. the tree describes that HMM module first visited its prefix state, which used the Dictionary module and Boundary module to generate its substring, where the Dictionary module used prefix re-, etc).

In the maximization phase, we use these collected description trees and we let them go through the probabilistic model from top: The top most module will analyze the trees and find out how often it e.g. transitioned from prefix state to itself or to root. It then takes the remainders of the trees and sends them to the lower layers, which again take their own information to update their own parameters and send the rest below, etc.

4.2 Morpheme→Morph model

We then created the morpheme→morph model. We model the morpheme realization simply by assigning each morpheme a list of morphs (strings) it could generate, altogether with probabilities of generation (Fig. 2 top).

To train the model we first need to create a candidate set of potential morphs for each morpheme - we take all substrings of original words. We then remove the substrings that do not co-occur with the morpheme sufficiently frequently to be reasonable candidates. Then we run the training procedure which finds the actual correct morphs: We train the probabilities of morpheme generating a given morph (Fig. 2 bottom). For each training sample we take sequence of morphemes, replace the root morpheme with a universal root generator and find the best mapping of morphemes to morphs so that the sequence of morphemes generates the original word. When we do this on a large amount of samples simultaneously, we can observe the probabilities that a given morpheme generates a given morph and we can use this information to update the morpheme generators - this can be interpreted as just another form of EM optimization and we ran it for multiple epochs. Once the training finished, we removed the morphs with low likelihood.

4.3 Final model

After joining the word→morphemes and morpheme→morph models we simply let the model find the most likely way of generating the word and give us the tree describing the generation (as discussed in the 4.1 chapter).

This generation tree is then analyzed and we look for the positions of the Boundary modules and for the Morpheme modules, which tells us the resulting segmentation.

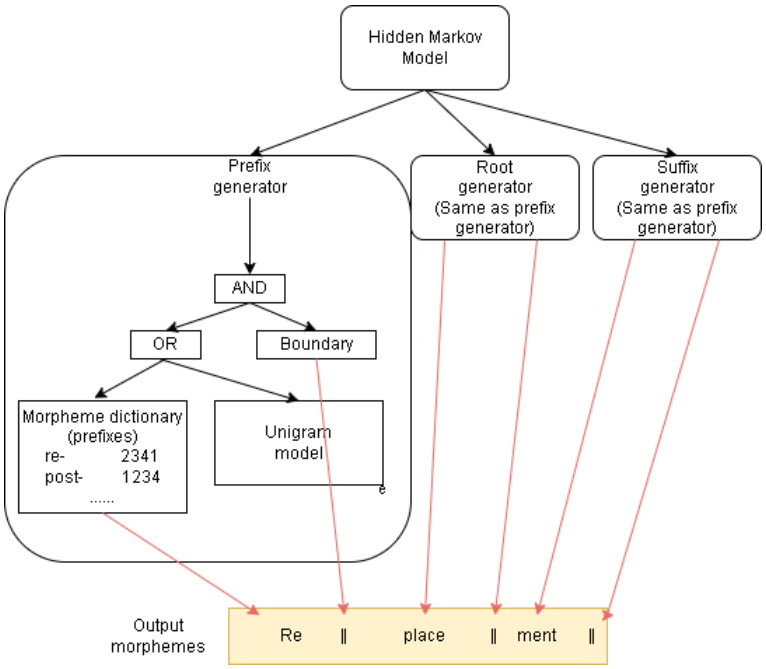

Figure 1: The architecture of word→morpheme model. Prefix-, Root-, and Suffix- generators are the same except for the dictionary. The uni-gram models generate string as combination of randomly selected letters (each letter has its probability). In the last phase of training, we will transform this model from generating morphemes to generating morphs: at first, morpheme dictionary just outputs the morpheme string for morpheme i (e.g. -s) with likelihood P[i]. After the transformation, it will output morpheme→morph model of morpheme i instead. This model will be then used to match the morphs (e.g. -s, -es, -en) in the input word. Boundary is a special sub-module that matches boundaries in the training phase and marks predicted boundaries in the inference phase

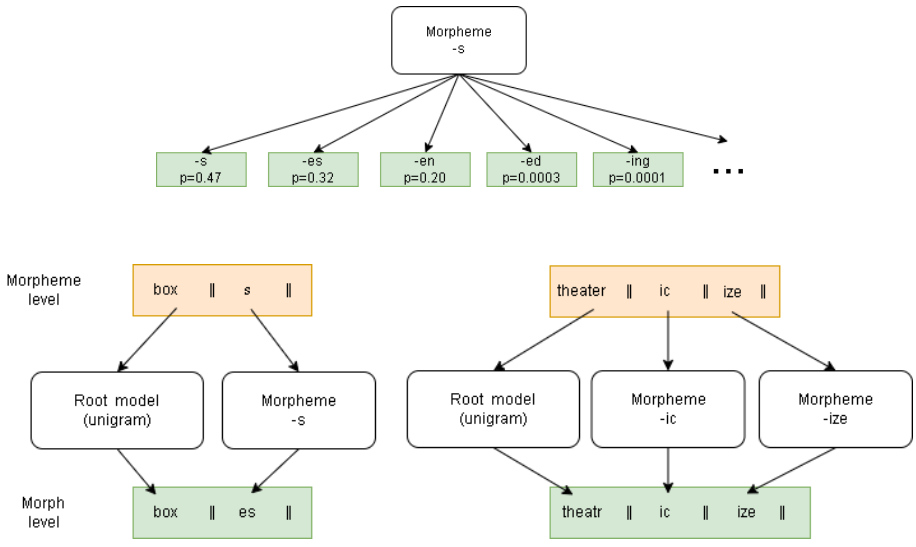

Figure 2: The architecture of morpheme→morph model (top) and the process of its training (bottom). The model describes generation of a morph as random choice among fixed candidates on the basis of trained probabilities. The training procedure works in such a way that it picks a word segmented to morphemes (red), uses it as a guideline for choice of morpheme models and looks for the best way how to use these morpheme models to generate the not segmented version of the word (green)

5 Results

The systems were evaluated via morpheme precision and recall. Precision is defined as the number of correctly predicted morphemes divided by total number of predicted morphemes. Recall is defined as the number of correctly predicted morphemes divided by total number of morphemes in the golden segmentation.

The following table summarizes our F-scores on the languages, as they were measured by the organizers of the Shared Task in (Batsuren et al., 2022).

Lang.	F1	Lang.	F1
Ces	64.65	Lat	91.39
Eng	65.43	Mon	57.82
Fra	46.20	Rus	50.55
Hun	72.64	Spa	43.39
Ita	33.44		

We can see that the model achieves relatively good results only on Latin (which was segmented to morphs) and not on other languages.

5.1 Error Analysis

This model was unable to achieve results comparable with the other approaches. We think that the main causes are following:

1. 1) Inability to capture the root changes, i.e. to transform the original root into its morpheme. (ENG: *emulations* = *emulate-ion-s*; SPA: *tricotemos* = *tricotar-emos*; ITA: *piastrellavamo* = *piastrellare-avamo*)

2. Missing context - the algorithm does not take surrounding letters into account when inserting a morph and it does not make use of joint probabilities of morphemes. Among other problems it also results in using a wrong morpheme for the generation of a morph (FRA: *recréerions* = *re-créer-erions* vs. *présidions* = *présider-ions*)

3. Morphemes with empty morph - probabilistic model of this type cannot generate morpheme from nothing (FRA: *agrémentant* = *agrér-ment-er-ant*). We would have to rely on joint probabilities of morphemes to derive it.

4. Under-segmentation - when we look at the results of the model on the Czech data (which are only segmented to morphs, not morphemes), then we notice that we discovered only 70% of boundaries between morphemes, but we have 95% precision on the boundary discovery. This was likely a consequence of removal of single letter morphemes from the model. Czech has tendency to use them frequently, as e.g. in *chyt-a-l-a*, or *bý-v-a-l-ý*, but they may cause problems with oversegmentation, as in *minim-al-iz-ova-t*, so it would be better to use a model that either groups the short morphs or incorporates joint probabilities of morphemes.

5. Root boundary detection - the model seems to have trouble detecting beginning and end of the root. When training the word→morphemes model we have observed that adding root dictionary (with roots extracted from the set of training morphemes) highly improves the segmentation accuracy. The problem is, that this dictionary cannot be directly transferred to the word→morphemes→morphs model, because root morphs in words are different from root morphemes in the dictionary, so some intermediate layer would be required.

6 Conclusion & Future Work

Our submission to the shared task on morphological segmentation was a modular probabilistic model trained via EM. The model has achieved poor results and the error analysis shows that a big amount of modifications will be needed in order to improve the results. Especially, the addition of more contextual information will be necessary. It also remains unclear how to handle differences between root morphs and root morphemes with this type of model.

7 Acknowledgement

This work was supported by the Grant No. START/HUM/010 of Grant schemes at Charles University (reg. No. CZ.02.2.69/0.0/0.0/19_073/0016935), the LINDAT/CLARIAH-CZ project of the Ministry of Education Youth and Sports of the Czech Republic (project LM2018101), and by the SVV project No. 260 575.

It was using language resources developed, stored, and distributed by the LINDAT/CLARIAH-CZ project.

References

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinović, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The sigmorphon 2022 shared task on morpheme segmentation. In *19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2004. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th meeting of the acl special interest group in computational phonology: Current themes in computational phonology and morphology*, pages 43–51.

John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Natural language engineering*, 12(4):353–371.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Morfessor em+ prune: Improved subword segmentation with expectation maximization and pruning. *arXiv preprint arXiv:2003.03131*.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio. Association for Computational Linguistics.