Automatically Tailoring Unsupervised Morphological Segmentation to the Language

Ramy Eskander[†] **Owen Rambow**[♯] **Smaranda Muresan**[†‡]
[†]Department of Computer Science, Columbia University
[‡]Data Science Institute, Columbia University
{rnd2110,smara}@columbia.edu
[♯]Elemental Cognition, Inc.
owenr@elementalcognition.com

Abstract

Morphological segmentation is beneficial for several natural language processing tasks dealing with large vocabularies. Unsupervised methods for morphological segmentation are essential for handling a diverse set of languages, including low-resource languages. Eskander et al. (2016) introduced a Language Independent Morphological Segmenter (LIMS) using Adaptor Grammars (AG) based on the best-on-average performing AG configuration. However, while LIMS worked best on average and outperforms other state-of-the-art unsupervised morphological segmentation approaches, it did not provide the optimal AG configuration for five out of the six languages. We propose two language-independent classifiers that enable the selection of the optimal or nearly-optimal configuration for the morphological segmentation of unseen languages.

1 Introduction

As natural language processing becomes more interested in many languages, including low-resource languages, unsupervised morphological segmentation remains an important area of study. For most of the languages of the world, we do not have morphologically annotated resources. However, many human language technologies profit from morphological segmentation, for example machine translation (Nguyen et al., 2010; Ataman et al., 2017) and speech recognition (Narasimhan et al., 2014).

In this paper, we build on previous work on unsupervised morphological segmentation using Adaptor Grammars (AGs) (Johnson, 2008; Sirts and Goldwater, 2013; Eskander et al., 2016), a type of nonparametric Bayesian models that generalize probabilistic context-free grammars (PCFGs) (Johnson et al., 2007), where the PCFG is typically a morphological grammar that specifies the word structure. Specifically, we extend the research proposed by Eskander et al. (2016), who investigate a large space of parameters when using Adaptor Grammars related to (i) the underlying context-free grammar and (ii) the use of a "Cascaded" system in which one grammar chooses affixes to be seeded into another in order to simulate the situation where scholar-knowledge is available. Their results on a development set of 6 languages (English, German, Finish, Turkish, Estonian and Zulu) show that the best performing AG-based configuration (grammar and learning setup) differ from language to language. For processing unseen languages, Eskander et al. (2016) proposed the Language-Independent Morphological Segmenter (LIMS) based on the best-on-average performing configuration when running leave-one-out cross validation on the development languages.

However, while LIMS works best on average and has been shown to outperform other state-of-the-art unsupervised morphological segmentation systems (Eskander et al., 2016), it is not the optimal configuration for any of the development languages except Zulu. Thus, in this paper we propose an approach to automatically select the optimal or nearly-optimal language-independent configuration for the morphological segmentation of unseen languages. We train two classifiers on the development languages used by Eskander et al. (2016) to make choices for unseen languages (Section 3). We show that we can choose the best parameter settings for the six development languages in a leave-one-out cross validation, and also on an unseen test language (Arabic).

2 Problem Definition and Dataset

Adaptor Grammars (AGs) have been used successfully for unsupervised morphological seg-

Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 78–83

https://doi.org/10.18653/v1/P17

Grammar	Main Representation	Compound	Morph	SubMorph	Segmentation Level
Morph+SM	Morph+	No	Yes	Yes	Morph
Simple	Prefix?+Stem+Suffix?	No	No	No	Prefix-Stem-Suffix
Simple+SM	Prefix?+Stem+Suffix?	No	No	Yes	Prefix-Stem-Suffix
PrStSu	Prefix+Stem+Suffix	No	Yes	No	Prefix-Stem-Suffix
PrStSu+SM	Prefix+Stem+Suffix	No	Yes	Yes	Prefix-Stem-Suffix
PrStSu+Co+SM	Prefix+Stem+Suffix	Yes	Yes	Yes	Prefix-Stem-Suffix
PrStSu2a+SM	Prefix?+(Stem+Suffix)	No	Yes	Yes	Prefix-Stem-Suffix
PrStSu2b+SM	(Prefix-Stem)+Suffix?	No	Yes	Yes	Prefix-Stem-Suffix
PrStSu2b++Co+SM	(Prefix-Stem)+Suffix?	Yes	Yes	Yes	Prefix-Stem-Suffix

Table 1: Grammar Representations. Compound = Upper level representation of the word as a sequence of compounds; Morph = Affix/Morph representation as a sequence of morphs. SubMorph (SM) = Lower level representation of characters as a sequence of sub-morphs. "+" denotes *one or more* and "?" denotes *optional*.

mentation (Johnson, 2008; Sirts and Goldwater, 2013; Eskander et al., 2016), which is the task of breaking down words in a language into a sequence of morphs. An AG model typically has two main components: a PCFG and an adaptor that adapts the probabilities assigned to individual subtrees in the grammar. For the task of morphological segmentation, a PCFG is typically a morphological grammar that specifies word structure. Given a list of input strings, AGs can learn latent tree structures.

Eskander et al. (2016) developed several AG models based on different underlying context-free grammars and learning settings, which we briefly introduce below.

Grammars. Eskander et al. (2016) introduce a set of 9 grammars (see Table 1) designed based on three dimensions: 1) how the grammar generates the prefix, stem and suffix (morph vs. tripartite), 2) the levels which are represented in nonterminals (e.g., compounds, morphs and sub-morphs) and 3) the levels at which the segmentation into output morphs is produced. For example, in the PrStSu+SM grammar a word is modeled as a prefix, a stem and a suffix, where the prefix and suffix are sequences of zero or more morphs, while a morph is a sequence of sub-morphs, and the segmentation is based on the prefix, suffix and stem level. The PrStSu2a+SM grammar is similar, but a word is modeled as a prefix and stem-suffix sequence, where the prefix is optional, and stem-suffix is either a stem or a stem and a suffix (see Eskander et al. (2016) for more details). Figure 1 shows the trees for segmenting the word *replayings* using the PrStSu+SM and PrStSu2a+SM grammars.

Learning Settings. Eskander et al. (2016) consider three learning settings: Standard (Std), Scholar-Seeded Knowledge (Sch) and Cascaded (Casc). In the Standard setting, no scholar knowledge is introduced in the grammars, while in the Scholar-Seeded Knowledge setting the grammars are augmented with scholar knowledge in the form of information about affixes gathered from grammar books (before learning happens). The Cascaded setting approximates the effect of scholar-seeded knowledge by first using a high-precision AG to derive a set of affixes and then insert those affixes into the grammars used in a second learning step.

Eskander et al. (2016) show that the segmentation performance differs significantly across the different grammars, learning settings and languages. For instance, the best performance for German is obtained by running the Standard PrStSu+SM configuration, while the Cascaded PrStSu2a+SM configuration produces the best segmentation for Finnish. That means, there is no setup that yields the optimal segmentation for all languages. For the processing of an unseen language (i.e., not part of the development), Eskander et al. (2016) recommend using the Cascaded PrStSu+SM configuration (referred to as LIMS: Language-Independent Morphological Segmenter), as it is the best-on-average performing one when running leave-one-out cross validation on the development languages.

Problem definition. While LIMS works best on average, it is not the optimal configuration for any of the development languages except Zulu. Thus, in this paper, we address the problem of automatically selecting the optimal or nearly-optimal language-independent (Standard or Cascaded) configuration for the morphological segmentation of unseen languages.

We use the 6 development languages used by Eskander et al. (2016) as well as Arabic as a fully unseen language. The data for English, German, Finnish, Turkish and Estonian is from Morpho Challenge[1], and the data for Zulu is from the Ukwabelana corpus (Spiegler et al., 2010). For the

[1] http://research.ics.aalto.fi/events/morphochallenge/

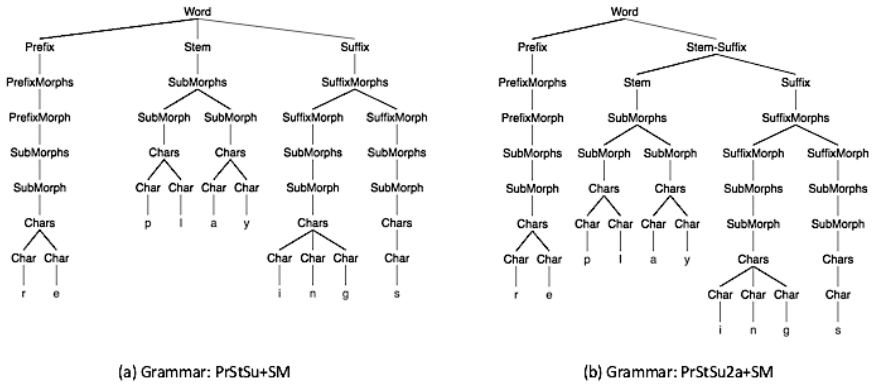

Figure 1: Grammar trees for the word *replayings*: (a) PrStSu+SM, (b) PrStSu2a+SM

Lang.	Source	TRAIN	DEV	TEST
English	Morpho Challenge	50,046	1,212	–
German	Morpho Challenge	50,086	540	–
Finnish	Morpho Challenge	49,909	1,494	–
Turkish	Morpho Challenge	49,765	1,531	–
Estonian	Morpho Challenge	49,909	1,500	–
Zulu	Ukwabelana	50,000	1,000	–
Arabic	PATB	50,000	–	1,000

Table 2: Data source and size information. TRAIN = training corpus, DEV = development corpus and TEST = test corpus.

unseen language we choose Arabic as it belongs to the Semitic family, while none of the development languages does. We obtain the Arabic data by randomly selecting 50K words from the PATB corpus (Maamourio et al., 2004). Table 2 lists the sources and sizes of our corpora.

3 Method

Since we have nine grammars to choose from (see Table 1) with two possible learning setting (Standard and Cascaded), for a total of 18 possible configurations, we restrict our pool of selection to the four configurations that yield the best-on-average performance across the development languages, namely Cascaded PrStSu+SM, Cascaded PrStSu2a+SM, Standard PrStSu+SM and Standard PrStSu2a+SM, with average EMMA F-scores (Spiegler and Monson, 2010) of 0.720, 0.695, 0.684 and 0.683, respectively (see Section 2 and Table 1 for grammar descriptions). EMMA stands for the Evaluation Metric for Morphological Analysis (Spiegler and Monson, 2010), and is a metric that has been shown to be particularly adequate for evaluating unsupervised methods for morphological segmentation and superior to the metric used in the Morpho Challenge competition series.

We use a supervised machine learning approach to select the best configuration. Since we only have six development languages, we split the classification task into two binary classification ones: Approach Classification (Standard (Std) vs. Cascaded (Casc)) and Grammar Classification (PrStSu+SM vs. PrStSu2a+SM), and run leave-one-out cross validation on the development languages for both tasks. Table 3 lists the best configurations and the gold class labels (for both Approach and Grammar) for the six development languages.

Language	**Best Configuration**	**Approach class**	**Grammar class**
English	Std PrStSu+SM	Std	PrStSu+SM
German	Std PrStSu+SM	Std	PrStSu+SM
Finnish	Casc PrStSu2a+SM	Casc	PrStSu2a+SM
Turkish	Std PrStSu+SM	Std	PrStSu+SM
Estonian	Casc PrStSu2a+SM	Casc	PrStSu2a+SM
Zulu	Casc PrStSu+SM	Casc	PrStSu+SM

Table 3: The best configurations and the gold class labels for both the Approach classification and Grammar classification for the six development languages.

3.1 Feature Generation

In order to generate morphological features for the classification tasks, we run a phase of AG segmentation using the Standard PrStSu+SM configuration, where we only run 50 optimization iterations (i.e., one tenth of the number of iterations in a complete segmentation process as

Feature ID	Description
F01	Average no. of simple affixes per word
F02	Average no. of simple prefixes per word
F03	Average no. of simple suffixes per word
F04	Average no. of characters per affix
F05	No. of distinct simple affixes
F06	No. of distinct simple prefixes
F07	No. of distinct simple suffixes
F08	Average no. of complex affixes per word
F09	Average no. of complex prefixes per word
F10	Average no. of complex suffixes per word
F11	Average no. of characters per affix
F12	No. of distinct complex affixes
F13	No. of distinct complex prefixes
F14	No. of distinct complex suffixes

Table 4: Classification features

Language	KNN	NB	RF
English	Std PrStSu+SM	Std PrStSu+SM	Std PrStSu+SM
German	Std PrStSu+SM	Std PrStSu+SM	Casc PrStSu+SM
Finnish	Casc PrStSu2a+SM	Casc PrStSu+SM (x)	Casc PrStSu2a+SM
Turkish	Std PrStSu+SM	Std PrStSu+SM	Std PrStSu+SM
Estonian	Casc PrStSu2a+SM	Casc PrStSu+SM (x)	Std PrStSu2a+SM (x)
Zulu	Casc PrStSu+SM	Casc PrStSu+SM	Casc PrStSu+SM
Accuracy	100.0%	66.7%	88.3%

Table 5: Overall system output. KNN = K-Nearest Neighbors, NB = Naive Bayes and RF = Random Forest. Wrong predictions are denoted by (x).

reported by Eskander et al. (2016)), as the purpose is to quickly generate morphological clues that help the classification rather than to obtain highly optimized segmentation. We choose this particular configuration due to its high efficiency across all languages in addition to its relatively small execution time. Upon generating the initial segmentation, we extract 14 morphological features for classification. The features are listed in Table 4. We only consider affixes that appear more than 10 times in the segmentation output, where a simple affix contains only one morpheme, while a complex affix contains one or more simple affixes.

3.2 Classification

We experiment with three classification methods; K-Nearest Neighbors (KNN), Naive Bayes (NB) and Random Forest (RF) for both the Approach (Std vs. Casc) and Grammar (PrStSu+SM vs. PrStSu2a+SM) classification tasks. We conduct the two classification tasks separately, and then we combine the outcome to obtain the best configuration.

In the training phase, we perform leave-one-out cross validation on the six development languages. In each of the six folds of the cross validation, we choose one language in turn as the test language. We use the training and development corpora listed in table 2 for training the models and evaluating the classifiers, respectively.

Table 5 shows the final system output after combining the outcomes from the Approach classification and Grammar Classification. KNN predicts the right configuration consistently, while NB picks the wrong grammars for Finnish and Estonian, and RF predicts the wrong approach and grammar for Estonian. Thus, the overall accuracies of KNN, NB and RF are 100%, 66.7% and 88.3%, respectively, which suggests using KNN for classification. So for an unseen language, we first run the Standard PrTuSu+SM configuration for 50 optimization iterations to obtain the morphological features. We then run the KNN classifier on those features in order to obtain the final AG configuration.

Studying the correlation between the morphological features and the output shows that features F14, F07, F11 and F03 in table 4, are the most significant ones for the selection of the best configuration. This illustrates the high reliance on information about suffixes as three out of the four features, namely F14, F07 and F03, are suffix-related.

4 Evaluation

We report results using the EMMA F-measure score (Spiegler and Monson, 2010).

Results on an unseen language. We evaluate our system on Arabic, a language that is not part of the development of the system. Arabic also belongs to the Semitic family, where none of the development languages does. For an unseen language, we first run the Standard PrStSu+SM configuration for 50 optimization iterations to obtain the morphological features. We then run the KNN classifier on those features in order to obtain the final AG configuration. Table 6 lists the EMMA F-scores for Arabic for all grammars in both the Standard and Cascaded setups. Our KNN classifier picks the Standard PrStSu+SM configuration, which yields the best segmentation among all the configurations with an EMMA F-score of 0.701.

Comparison with existing unsupervised approaches. Table 7 compares the performance of the selected configurations of our system (Table 5) to three other systems; Morfessor (Creutz and Lagus, 2007), MorphoChain (Narasimhan et al., 2015) and LIMS (Eskander et al., 2016) (where the cascaded PrStSu+SM configuration is

Grammar	Standard	Cascaded
Morph+SM	0.647	0.642
Simple	0.651	0.593
Simple+SM	0.680	0.631
PrStSu	0.642	0.646
PrStSu+SM	**0.701**	0.692
PrStSu+Co+SM	0.648	0.628
PrStSu2a+SM	0.676	0.682
PrStSu2b+SM	0.682	0.688
PrStSu2b+Co+SM	0.532	0.532

Table 6: Adaptor-grammar results (Emma F-scores) for the Standard and Cascaded setups for Arabic. Boldface indicates the best configuration and the choice of our system.

Grammar	Morfessor	MorphoChain	LIMS	Ours	Best
English	0.805	0.746	0.809	0.821	0.826
German	0.740	0.625	0.777	0.790	0.790
Finnish	0.675	0.621	0.727	0.733	0.733
Turkish	0.551	0.551	0.591	0.647	0.647
Zulu	0.414	0.390	0.611	0.611	0.611
Estonian	0.779	0.679	0.805	0.828	0.847
Arabic	0.779	0.751	0.682	0.701	0.701
Avg.	0.678	0.623	0.715	0.733	0.736

Table 7: The performance of our system (Ours) compared to Morfessor, MorphoChain, LIMS and an upper-bound system (Best), using EMMA F-scores.

chosen). Our system has EMMA F-score error reductions of 17.1%, 29.2% and 6.3% over Morfessor, MorphoChain[2] and LIMS, respectively, on average across the development languages and Arabic. It is also only 0.003 of average EMMA F-score behind an oracle system, where the best configuration is always selected (indicated as *Best*). We are not able to compare versus the system presented by Wang et al. (2016) as neither their system nor their data is currently available.

5 Related Work

The first work that utilizes AGs for unsupervised morphological segmentation is introduced by Johnson (2008), while Sirts and Goldwater (2013) propose minimally supervised AG models of different tree structures for morphological segmentation. The most recent work on using AGs for morphological segmentation is proposed by Eskander et al. (2016), where they experiment with several AG models based on different underlying grammars and learning settings. They also research the use of scholar knowledge seeded in the grammar trees. This knowledge could be gathered from grammar books or automatically generated via bootstrapping. This paper extends their work by proposing a machine learning approach to select the best language-independent model for each language.

In addition to the use of AGs, several models have been successfully used for unsupervised morphological segmentation such as generative probabilistic models (utilized by Morfessor (Creutz and Lagus, 2007)), and log-linear models using contextual and global features (Poon et al., 2009). Narasimhan et al. (2015) use a discriminative model for unsupervised morphological segmentation that integrates orthographic and semantic properties of words. The model learns morphological chains, where a chain extends a base form available in the lexicon.

Another recent notable work is introduced by Wang et al. (2016), who use neural networks for unsupervised segmentation, where they build LSTM (Hochreiter and Schmidhuber, 1997) architectures to learn word structures in order to predict morphological boundaries. Another variation of the this approach is presented by Yang et al. (2017), where they use partial-word information as character bigram embeddings and evaluate their work on Chinese.

6 Conclusion and Future Work

We have shown that our language-independent classifiers improve the state-of-the-art unsupervised morphological segmentation proposed by Eskander et al. (2016) by making choices that optimize for a given language, rather than choosing parameters for all languages based on averages on the development languages.

In future work, we plan to conduct an extrinsic evaluation on tasks that could benefit from morphological segmentation such as machine translation, information retrieval and summarization. We also plan to optimize the segmentation models for those specific tasks.

Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[2]Since MorphoChain expects large corpora in order to learn the morphological chains, it does not perform well on the small corpora we use in our setup, where we experiment with real conditions of low-resource languages.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. 108.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34.

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016. Extending the use of adaptor grammars for unsupervised morphological segmentation of unseen languages. In *Proceedings of he Twenty-Sixth International Conference on Computational Linguistics (COLING)*, Osaka, Japan.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mark Johnson. 2008. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio. Association for Computational Linguistics.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: a framework for specifying compositional nonparametric bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA. MIT Press.

Mohamed Maamourio, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. Arabic treebank: Building a large-scale annotated arabic corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. In *Twelfth AAAI Conference on Artificial Intelligence*.

Karthik Narasimhan, Damianos Karakos, Richard M. Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *EMNLP*.

ThuyLinh Nguyen, Stephan Vogel, and Noah A. Smith. 2010. Nonparametric word segmentation for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 815–823, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Boulder, Colorado. Association for Computational Linguistics.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1(May):231–242.

Sebastian Spiegler and Christian Monson. 2010. Emma: A novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1029–1037, Beijing, China. Coling 2010 Organizing Committee.

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. 2010. Ukwabelana - an open-source morphological zulu corpus. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1020–1028, Beijing, China. Coling 2010 Organizing Committee.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological segmentation with window LSTM neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*, Vancouver, Canada.