Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 7112–7122
Marseille, 11–16 May 2020

MorphAGram: Evaluation and Framework for Unsupervised Morphological Segmentation

Ramy Eskander[†], **Francesca Callejas**[†], **Elizabeth Nichols**[††], **Judith Klavans**[†††], **Smaranda Muresan**[†]
Columbia University, USA[†], Barnard College, USA[††], University of Maryland, USA[†††]
{rnd2110,ffc2108,smara}@columbia.edu[†], en2433@barnard.edu[††], jklavans@umd.edu[†††]

Abstract

Computational morphological segmentation has been an active research topic for decades as it is beneficial for many natural language processing tasks. With the high cost of manually labeling data for morphology and the increasing interest in low-resource languages, unsupervised morphological segmentation has become essential for processing a typologically diverse set of languages, whether high-resource or low-resource. In this paper, we present and release MorphAGram, a publicly available framework for unsupervised morphological segmentation that uses Adaptor Grammars (AG) and is based on the work presented by Eskander et al. (2016). We conduct an extensive quantitative and qualitative evaluation of this framework on 12 languages and show that the framework achieves state-of-the-art results across languages of different typologies (from fusional to polysynthetic and from high-resource to low-resource).

Keywords: Unsupervised Morphological Segmentation Framework, Low-Resource Languages, Qualitative and Qualitative Evaluation, Adaptor Grammars, Language Typology

1. Introduction

Many natural language processing tasks profit from morphological segmentation, for example machine translation (Nguyen et al., 2010; Ataman et al., 2017) and speech recognition (Narasimhan et al., 2014). Many of the languages of the world are low-resource and/or endangered, where they lack adequate morphologically annotated resources. Thus, open-source unsupervised morphological segmentation frameworks could be an important resource for the computational linguistics community. In addition, we argue that frameworks that enable the use of linguistic knowledge to guide the learning process could be particularly beneficial when working on low-resource or endangered languages, where even unsegmentated data might be minimal.

We present MorphAGram [1], an publicly available framework for unsupervised morphological segmentation based on the approach proposed by Eskander et al. (2016), that uses Adaptor Grammars. Formal grammars, and particularly Context-Free Grammars (CFGs), are a keystone of linguistic description and provide a model for the structural description of linguistic objects. Probabilistic CFGs (PCFGs) extend this model by associating a probability to each context-free rewrite rule. Adaptor grammars (AGs) (Johnson et al., 2007) weaken the independence assumptions of PCFGs by inserting additional stochastic processes called adaptors into the procedure for generating structures. Introducing dependencies among the applications of rewrite rules extends the set of distributions over linguistic structures that can be characterized by a grammar, better matching the occurrences of trees and sub-trees observed in actual corpora. AGs define a framework to implement Bayesian nonparametric learning of grammars and are usually trained in an unsupervised manner using sampling techniques. AGs have been used successfully for unsupervised morphological segmentation, where a grammar is a morphological grammar that specifies word structure (Sirts and Goldwater, 2013; Eskander et al., 2016; Eskander et al., 2018; Eskander et al., 2019). AGs have been also applied to other NLP applications such as word segmentation (Johnson, 2008a; Johnson, 2008c; Johnson and Demuth, 2010), named-entity clustering (Elsner et al., 2009), transliteration of names (Huang et al., 2011) and native-language identification (Wong et al., 2012).

In this paper, we release MorphAGram, a publicly available framework for unsupervised morphological segmentation. The framework is also suitable for semi-supervised learning setups where it allows linguistic knowledge to be specified at two levels: designing the grammars and using scholar-seeded knowledge in terms of known affixes (Section 3). We conduct an extensive quantitative and qualitative evaluation of this framework (Section 4) for a set of 12 languages across a language typology continuum (Section 2), namely English, German, Finnish, Estonian, Georgian, Turkish, Arabic, Zulu, Mexicanero, Nahuatl, Wixarika and Yorem Nokki. Our results show state-of-the-art results for this framework, and showcase that for some languages using linguistic knowledge in terms of known affixes helps, even when the grammars are language-independent. Both the code and the grammars are released with the framework.

2. Language Typology and Morphological Analysis

The type of language impacts the way languages should be analyzed since wide-ranging cross-linguistic typological differences exist between languages (Comrie, 1993); and one of these parameters is based on morphological properties and usage of different affixation processes. The broadest distinction among languages is whether or not affixation is allowed at all, or if every word must be a single morpheme. Although language typology forms a framework for morphological analysis, it is important to remember that a typology involves a logical continuum along which languages can differ synchronically and move diachronically. Thus, most languages are mixtures of typological distinc-

[1] https://github.com/rnd2110/MorphAGram

tions with a primary type as the one to be chosen as most salient. This paper addresses languages from several typological continua and demonstrates that the unsupervised Adapter-Grammar method performs well regardless of typology. The challenges of segmentation are directly related to the type of affixation and cliticization found in each language (Klavans, 2018).

Isolating Languages In isolating languages, every word must be a single morpheme (no affixation). These are *isolating* and fully analytic languages. This makes segmentation more clear in this language type.

Synthetic Languages In contrast to isolating languages are *synthetic* languages, which allow affixation; words may (though are not required to) include two or more morphemes. These languages have bound morphemes, meaning they must be attached to another word (whereas analytic languages almost exclusively only have free morphemes). Synthetic languages include three subcategories: *agglutinative*, *fusional*, and *polysynthetic*. An agglutinating language (e.g., Turkish or Finnish) is one in which word forms can be easily and clearly segmented into individual morphs, each of which represents a single grammatical category. In this case, morphological segmentation can be achieved in a relatively straightforward way since individual components are easily recognizable as units even though the "word" may appear to be long and complex. This is generally just a matter of many morphemes joined together. Moving along the synthetic continuum are fusional languages, where bound morphemes often blend two or more underlying functions into one, and these are not easily decomposable. Unlike agglutination, there may be no one-to-one correspondence between specific word segments and particular grammatical categories. For example, the Latin suffix *-is* represents the combination of categories "singular" and "genitive" in the word form *hominis* "of the man", but one part of the suffix cannot be assigned to "singular" and another to "genitive," and *-is* is only one of many suffixes that, in different classes (or declensions) of words, represents the combination of "singular" and "genitive". At the other end of the extreme are polysynthetic languages, where many morphemes fuse into one unit, known as the "word" but also often representing an entire sentence, and replete with verbs, nouns and clauses. These languages are difficult for computational systems (and for non-native speakers) to deconstruct and analyze due to the high level of ambiguity in segmentation and to the lack of one-to-one mapping.

Languages Considered for Analysis In this paper, we consider 12 languages that are spread across the typology spectrum and for which morphologically segmented datasets are available for evaluation. These languages are:

- English: fusional, mildly synthetic
- German: fusional, more synthetic
- Finnish: agglutinative, more synthetic
- Estonian: agglutinative, more synthetic
- Georgian: agglutinative, mildly fusional
- Turkish: agglutinative, more synthetic
- Arabic (MSA): fusional, less synthetic
- Zulu: agglutinative, mildly fusional
- Mexicanero: polysynthetic
- Nahuatl: polysynthetic
- Wixarika: polysynthetic
- Yorem Nokki: polysynthetic

3. Framework

As pointed out earlier, Adaptor Grammars (AGs) are nonparameteric Bayesian models that generalize Probabilistic Context Free Grammars (PCFGs) (Johnson et al., 2007). An AG is composed of two main components: a PCFG and an adaptor that adapts the probabilities of the sub-trees and acts as a caching model. The adaptor is based on the Pitman-Yor process (Pitman, 1995), where the posterior probability of a subtree is kept proportional to the number of times that subtree is utilized given the input data. Markov Chain Monte Carlo sampling (MCMC) (Andrieu et al., 2003) is then used to infer the probabilities of the production rules of the grammar and all the hyperparameters of the model. The definition of the grammar relies on the underlying task. In the case of morphological segmentation, the grammar specifies the word structure in the underlying language.

We next describe the different parts of our segmentation framework, MorphAGram, that is based on Adaptor Grammars.

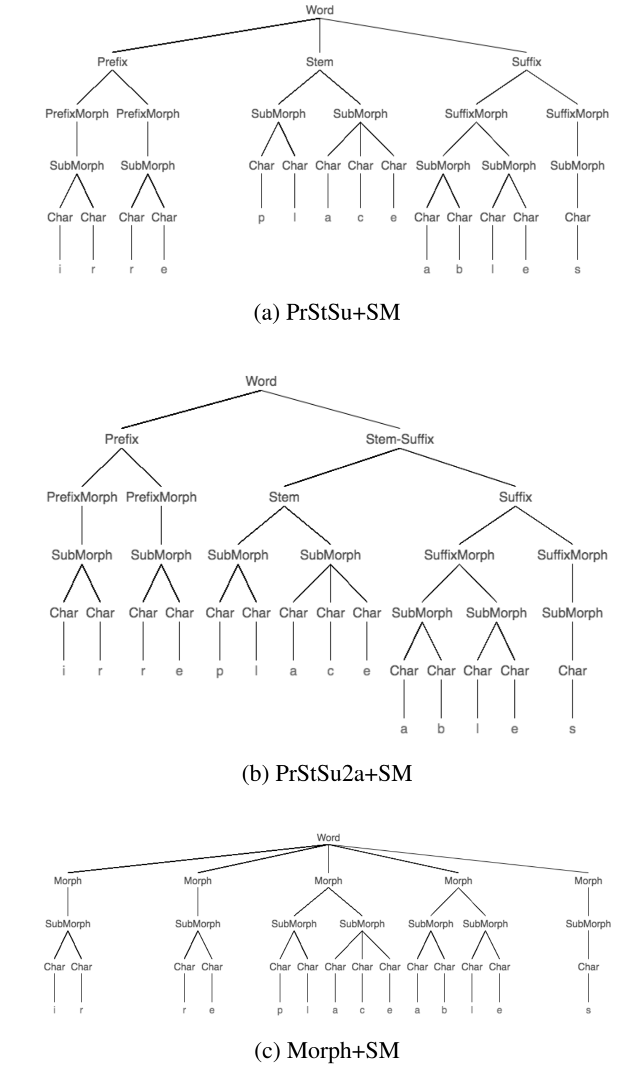
3.1. Defining the Grammar

The first step in learning morphological segmentation using Adaptor Grammars is to define the grammar; nonterminals, terminals and production rules. Eskander et al. (2016) describe a list of nine grammars that specify word structures, where the construction of the grammar relies on three main dimensions:

- **Word Modeling**: A word can be modeled as a sequence of generic morphs or as a sequence of prefixes, a stem and a sequence of suffixes.
- **Level of Abstraction**: Basic non-terminals can be combined into more complex categories, e.g., *Compounds*, or split into smaller ones, e.g., *SubMorphs*.
- **Segmentation Boundaries**: This defines the nonterminals that incur the splits in the final segmentation output. For example, a word can be segmented on the level of complex affixes, e.g., *re+play+ings*, or simple ones, e.g., *re+play+ing+s*.

Figure 1 shows the grammar trees of three different grammars: *PrStSu+SM*, *PrStSu2a+SM* and *Morph+SM*, representing the English word *irreplaceables*. A *PrStSu* grammar is a grammar where a word is modeled as a sequence of prefixes, a stem and a sequence of suffixes, where *Pr*, *St* and *Su* refer to *Prefix*, *Stem* and *Suffix*, respectively. The addition of *SM* means the basic components are split into sub-morphs. The term *2a* refers to a variation of the *PrStSu+SM* grammar where the stem and the list of the suffixes are combined into a parent category *StemSuffixes*. On the other hand, the *Morph+SM* grammar, proposed by Sirts and Goldwater (2013), has words modeled as a sequence of morphs that are composed of sub-morphs. For more details about the specifications of these grammars and other grammars, see Eskander et al. (2016).

In addition to defining the main grammar, each production rule has to be associated with three parameters; ϑ, *a* and *b*, where ϑ is the probability of the rule in the generator, while *a* and *b* are the parameters of the Pitman-Yor process (Pitman et al., 1997). If not specified, the parameters are sampled by the trainer, or they can be set to default values prior to running the learner. Setting *a* to one means the underlying non-terminal is not adapted and is sampled by the general Pitman-Yor process, while setting *a* to zero means the adaptor of the non-terminal is a Dirichlet process (Ishwaran and James, 2003) with the concentration parameter *b*. When a non-terminal is adapted, each sub-tree that can be generated using the initial rule of that non-terminal is considered as a potential rule in the grammar. Otherwise, the non-terminal expands as in a regular PCFG. For more details, see Johnson et al. (2007) and Johnson (2008b).



(a) PrStSu+SM

(b) PrStSu2a+SM

(c) Morph+SM

Figure 1: The representation of the English word *irreplaceables* segmented using three different grammars: PrStSu+SM, PrStSu2a+SM and Morph+SM

3.2. Training the Model

Inputs The two main inputs to the learner are the grammar, along with the adaptation information, and the vocabulary of the language we want to learn the segmentation for. The vocabulary is represented as a unique list of unsegmented words. If the size of the vocabulary is relatively large (e.g., more than 50K words), we recommend providing only the most frequent words in the underlying language. We then can obtain the segmentation of the remaining words in an inductive-learning manner. The details of text segmentation are discussed in Subsection 3.3.

Learning Settings Eskander et al. (2016) define three learning settings: Standard, Scholar-seeded and Cascaded.

- **Standard**: The Standard setting is language-independent, where a grammar does not have any language-specific production rules, and the learning is fully unsupervised. Figure 2 shows the input of the PrStSu+SM grammar in the standard mode, where ^^^ and *$$$* indicate the beginning and end of words, respectively.

- **Scholar-seeded**: In the case where some linguistic knowledge is available, e.g., a list of morphemes, this knowledge can be seeded into the grammar trees as additional production rules, allowing for a semi-supervised learning setup. Figure 3 shows the input of the PrStSu+SM grammar in the scholar-seeded mode, where a sample of English prefixes and suffixes are added.

- **Cascaded**: The cascaded setting approximates the effect of the scholar-seeded setting in a language-independent setup. This is done by first obtaining a list of morphemes from a segmentation model that is trained on a high-precision grammar, and then seeding those morphemes into another grammar. In this setup, both the standard grammar and the segmentation output of another grammar are provided. The system then extracts the top morphemes, typically affixes, from the segmentation output and seeds them into the standard grammar prior to running the learner.

It is worth noting that a production rule that is not preceded by parameters in Figure 2 and Figure 3 has a default zero value for the *a* parameter in the Pitman-Yor process, which means the rule is adapted. On the other hand, those rules preceded by "1 1" are not adapted, where the first number represents the value of the probability of the rule in the generator, ϑ, and the second number is the value of the *a* parameter in the Pitman-Yor process.

Inference The Markov Chain Monte Carlo (MCMC) approach is then used to infer the posterior distribution over the trees using a component-wise Metropolis-Hastings sampler. This also infers all the hyperparameters of the model, including the PCFG probabilities in the base distribution and the hyperparameters of the Pitman-Yor process. For comprehensive details about the inference algorithm and software implementation, see Johnson et al. (2007).

```
1 1 Word --> Prefix Stem Suffix

Prefix --> ^^^
Prefix --> ^^^ PrefixMorphs
1 1 PrefixMorphs --> PrefixMorph PrefixMorphs
1 1 PrefixMorphs --> PrefixMorph
PrefixMorph --> SubMorphs

Stem --> SubMorphs

Suffix --> $$$
Suffix --> SuffixMorphs $$$
1 1 SuffixMorphs --> SuffixMorph SuffixMorphs
1 1 SuffixMorphs --> SuffixMorph
SuffixMorph --> SubMorphs

1 1 SubMorphs --> SubMorph SubMorphs
1 1 SubMorphs --> SubMorph
SubMorph --> Chars
1 1 Chars --> Char
1 1 Chars --> Char Chars
```

Figure 2: The standard PrStSu+SM grammar

```
1 1 Word --> Prefix Stem Suffix
Prefix --> ^^^
Prefix --> ^^^ PrefixMorphs
1 1 PrefixMorphs --> PrefixMorph PrefixMorphs
1 1 PrefixMorphs --> PrefixMorph
PrefixMorph --> SubMorphs
Stem --> SubMorphs
Suffix --> $$$
Suffix --> SuffixMorphs $$$
1 1 SuffixMorphs --> SuffixMorph SuffixMorphs
1 1 SuffixMorphs --> SuffixMorph
SuffixMorph --> SubMorphs
1 1 SubMorphs --> SubMorph SubMorphs
1 1 SubMorphs --> SubMorph
SubMorph --> Chars
1 1 Chars --> Char
1 1 Chars --> Char Chars

1 1 PrefixMorph --> (a) (n) (t) (i)
1 1 PrefixMorph --> (s) (e) (m) (i)
1 1 PrefixMorph --> (e) (x) (t) (r) (a)
1 1 PrefixMorph --> (d) (i) (s)
1 1 PrefixMorph --> (n) (o) (n)
1 1 PrefixMorph --> (p) (r) (e)
1 1 SuffixMorph --> (e) (r)
1 1 SuffixMorph --> (e) (d)
1 1 SuffixMorph --> (s)
1 1 SuffixMorph --> (i) (n) (g)
1 1 SuffixMorph --> (i) (s) (t)
1 1 SuffixMorph --> (l) (e) (s) (s)
1 1 SuffixMorph --> (n) (e) (s) (s)
1 1 SuffixMorph --> (m) (e) (n) (t)
1 1 SuffixMorph --> (t) (i) (o) (n)
1 1 SuffixMorph --> (’) (s)
```

Figure 3: The scholar-Seeded PrStSu+SM grammar for English

3.3. Text Segmentation

The output of the inference algorithm includes the PCFG with the inferred hyperparameters and the generated subtrees that correspond to the adapted nonterminals in addition to the segmentation output of the input vocabulary. Text segmentation can then be performed in two different modes; transductive and inductive. In the transductive mode, the word should be present in the vocabulary list provided to the learner, where the segmentation output serves as a segmentation lookup. In contrast, the inductive mode is suitable for words that were not processed by the learner, where the segmentation is performed by parsing the input words given the output PCFG using a PCFG parsing algorithm such as CKY.

4. Evaluation and Results

In this section, we evaluate our morphological-segmentation framework, qualitatively and analytically. To review, we process the languages outlined in Section 2, where the details about their morphological characteristics are discussed. We start by describing our datasets, evaluation setups and evaluation metrics. We then show the performance of our segmentation framework compared to state-of-the-art baselines, in addition to the correlation between the size of the training set and segmentation quality. Finally, we conclude with analyzing the common errors produced by our models.

4.1. Data

The data for English, German, Finnish and Turkish is from the Morpho Challenge competition [2] (MC2010) (Kurimo et al., 2010), where we select the most frequent 50,000 words for training after filtering out those words that have foreign letters. In addition, the development sets are collected from all the years of the competition, where we filter out the German words in which the stem receives transformation.

The Estonian training and development sets are the ones used by Sirts and Goldwater (2013) [3], where we filter out words containing foreign letters. The data is based on the Sega corpus [4], where the gold segmentation in the development set is collected from the Estonian Morphologically Disambiguated Corpus [5].

The training data for Georgian is based on the most common 50,000 words in the Georgian Wikipedia, while the gold annotations in the development set are manually annotated in house.

The Arabic data is collected from the most frequent 50,000 words in the Arabic PATB Corpus (Maamourio et al., 2004), where the words in the development set are randomly selected. Similarily, the Zulu data is collected from the Ukwabelana corpus (Spiegler et al., 2010).

For Mexicanero, Nahuatl, Wixarika and Yorem Nokki, we use the data released by Kann et al. (2018) after cleaning up those words that are not white-space tokenized or containing foreign letters.

In all languages, we train our models using the training sets (*TRAIN*) in an unsupervised manner, while we use the development sets (*DEV*) for evaluation. We also use the test sets (*TEST*) of the polysynthetic languages as additional evaluation sets.

Table 1 reports the source of the data and the sizes of *TRAIN*, *DEV* and *TEST* per language. We also release all the datasets we use in this paper [6].

Table 2 reports morpheme-level statistics for the different languages we are experimenting with, based on *DEV*. The

[2] http://research.ics.aalto.fi/events/morphochallenge2010/datasets.shtml

[3] through contacting the authors directly

[4] https://keeleressursid.ee/et/196-segakorpus-eesti-ekspress

[5] https://www.cl.ut.ee/korpused/morfkorpus/

[6] https://github.com/rnd2110/MorphAGram

Language	Source	TRAIN	DEV	TEST
English	Morpho Challenge	50,000	1,212	NA
German	Morpho Challenge	50,000	556	NA
Finnish	Morpho Challenge	50,000	1,494	NA
Estonian	Sgea Corpus	49,621	1,492	NA
Georgian	Morpho Challenge	50,000	1,000	NA
Turkish	Morpho Challenge	50,000	1,531	NA
Arabic	PATB	50,000	1,000	NA
Zulu	Ukwabelana Corpus	50,000	1,000	NA
Mexicanero	Kann et al. (2018)	424	106	353
Nahuatl	Kann et al. (2018)	535	133	444
Wixarika	Kann et al. (2018)	664	166	550
Yorem Nokki	Kann et al. (2018)	509	126	421

Table 1: Data source and number of words in the training, development and test sets per language

second column lists the average morpheme length in the corresponding language, while the third column shows the average number of morphemes per word. The maximum number of morphemes in a word is reported in the fourth column. Finally, the last column lists the degree of ambiguity, which we define as:

$$1 - 2 \times \left| 0.5 - \frac{\sum_{i=1}^{n} \frac{N_i}{M_i}}{n} \right|$$

Where:

- n is the number of morphemes.
- N is the number of occurrences of the sequence of characters constituting the ith morpheme.
- M_i is the number of occurrences of the ith morpheme.

Language	Ave L(M)	Ave M/W	Max M/W	Ambiguity
English	5.30	2.39	6	0.48
German	5.16	2.94	8	0.43
Finnish	5.74	3.48	9	0.53
Estonian	5.63	1.93	7	0.48
Georgian	4.18	2.99	8	0.65
Turkish	4.61	3.30	8	0.65
Arabic	4.29	2.25	5	0.27
Zulu	4.60	3.96	9	0.64
Mexicanero	4.39	1.93	7	0.59
Nahuatl	4.58	2.31	6	0.69
Wixarika	4.16	3.30	10	0.75
Mayo	4.01	2.24	5	0.59

Table 2: Language statistics based on the development sets. M=Morpheme, W=Word, L(M)=Length of Morpheme.

4.2. Evaluation Setup

We evaluate two models per language: 1) the best language-independent (standard/cascaded) setup, denoted as AG-LI, and 2) the best scholar-seeded setup, denoted as AG-SS. We use the system proposed by Eskander et al. (2016) to obtain the best language-independent setups, which we report on in Table 3.

Language	Best AG-LI	Best AG-SS
English	Std. PrStSu+SM	Sch. PrStSu+SM
German	Std. PrStSu+SM	Sch. PrStSu+SM
Finnish	Casc. PrStSu+SM	Sch. PrStSu+SM
Estonian	Casc. PrStSu+SM	Sch. PrStSu+SM
Georgian	Casc. PrStSu+SM	Sch. PrStSu+SM
Turkish	Std. PrStSu+SM	Sch. PrStSu2a+SM
Arabic	Std. PrStSu+SM	Sch. PrStSu2a+SM
Zulu	Casc. PrStSu+SM	Sch. PrStSu+SM
Mexicanero	Std. PrStSu+SM	Sch. PrStSu+SM
Nahuatl	Std. PrStSu+SM	Sch. PrStSu+SM
Wixarika	Std. PrStSu+SM	Sch. PrStSu+SM
Yorem Nokki	Std. PrStSu+SM	Sch. PrStSu+SM

Table 3: The best language-independent (standard/cascaded) setup (AG-LI) and the best scholar-seeded setup (AG-SS) per language. Std.=Standard, Casc.=Cascaded, and Sch.=Scholar-Seeded

We conduct the evaluation in a transductive learning scenario, where the unsegmented test words are included in our training set, which is common in the evaluation of unsupervised morphological segmentation (Poon et al., 2009; Sirts and Goldwater, 2013; Narasimhan et al., 2015; Eskander et al., 2016). However, we do not see gains in the performance when using the inductive learning approach instead, where the unsegmented test words are separate from the training set.

We run the learners for 500 iterations for all languages, and we compute the results as the average of five runs since the samplers are non-deterministic. No annealing is used as it does not improve the results, and all parameters are automatically inferred.

4.3. Evaluation Metrics

We evaluate the performance of our morphological-segmentation framework using two metrics: Boundary Precision and Recall (BPR) and EMMA-2 (Virpioja et al., 2011). BPR is the classical evaluation method for morphological segmentation, where the boundaries in the proposed segmentation are compared to the boundaries in the reference. In contrast, EMMA-2 is based on matching the morphemes, and is a variation of EMMA (Spiegler and Monson, 2010). In EMMA, each proposed morpheme is matched to each morpheme in the gold segmentation through one-to-one mappings. However, EMMA-2 allows for shorter computation times as it replaces the one-to-one assignment problem in EMMA by two many-to-one assignment problems, where two or more proposed morphemes can be mapped to one reference morpheme. EMMA-2 also results in higher precision and recall as it tolerates failing to join two allomorphs or to distinguish between identical syncretic morphemes.

4.4. Baselines

We evaluate our system versus two state-of-the-art baselines: Morfessor (Creutz and Lagus, 2007) and MorphoChain (Narasimhan et al., 2014). Morfessor is a commonly used framework for unsupervised and semi-supervised morphological segmentation and is publicly available for free [7]. Morfessor utilizes the Minimum Description Length (MDL) concept for the selection of the optimal segmentation for both the input vocabulary and the segmentation lexicon. It also uses an HMM model that encodes the positional information of the morphemes. MorphoChain is another publicly available system for unsupervised morphological segmentation [8]. In MorphoChain, words are modeled as morphological chains, where a chain is a sequence of words that starts with a base word (parent) and ends up with a morphological variant. It uses a log-linear discriminative model to predict the parent of a given word, and uses the transformations in the underlying chain to derive the segmentation.

4.5. System Performance

Table 4 reports the performance of the best language-independent model (AG-LI) and the best scholar-seeded model (AG-SS) versus Morfessor and MorphoChain, for each language when tested on DEV, using the BPR and EMMA-2 metrics.

When using the BPR metric, our systems do constantly better than Morfessor and MorphoChain on all languages, where the AG-LI model decreases the errors produced by Morfessor and MorphoChain by 26.0% and 38.0%, respectively, on average across all languages, while the AG-SS model outperforms the AG-LI model by an average F1-score of 1.7%. We obtain the same patterns when applying the EMMA-2 metric, where our AG-LI model outperforms Morfessor and MorphoChain on all languages, but with a smaller gap than that of the BPR metric.

It is worth noting that models that tend to under-segment achieve significantly better EMMA-2 scores as opposed to the BPR ones, which is due to the one-to-many mappings in EMMA-2. This is one of the main reasons why system rankings may differ depending on the evaluation metric. An example is the considerable increase in the F1-score from 49.3%, when using BPR, to 81.1%, when using EMMA-2, when evaluating MorphoChain on Yorem Nokki, where MorphoChain does under-segmentation with 100% precisions and low recalls when detecting common affixes such as *m*, *ne po* and *su*.

Table 4 reports the performance of our systems compared to Morfessor and MorphoChain for the polysynthetic languages when tested on *TEST* . The AG-LI and AG-SS models achieve the best results on all languages, where the AG-LI model achieves absolute average F1-score increases of 25.4% and 46.2% over Morfessor and MorphoChain, respectively, when using the BPR metric.

4.6. Learning Curves

We examine the performance of the AG-LI and AG-SS models on German, Turkish and Arabic when training on different sizes: 500, 1K, 5K, 10K, 20K, 30K, 40K and 50K. The learning curves are reported in Figure 4.6.

The learning behavior for Arabic meets the expectations, where the performance consistently increases by adding more training data, while the AG-SS model always outperforms the AG-LI model across the different training sizes. In contrast, augmenting the training data for German and Turkish sometimes results in performance drops, despite the overall upward learning patterns. One explanation is that some data points might confuse the learner when added, leading to hyperparameters that are less efficient.

It is noteworthy to mention that the performance of the system on German and Turkish when only using 5,000 and 10,000 training words, respectively, outperforms the performance of the baseline systems, Morfessor and MorphoChain, when they utilize the full training set. This is because the system learns well from a small amount of data, which is the case when learning the segmentation for the polysynthetic languages as well.

4.7. Error Analysis

Table 6 shows some examples of correctly and incorrectly segmented words by our models, where incorrect ones are marked in red and italic characters, for seven languages: German, Georgian, Turkish, Arabic, Zulu, Wixarika and Yorem Nokki.

German Our models recognize the affixes *an ab* and *auf* with a high average recall of 94.8%. While *an*, *ab* auf have low relative frequencies in the data, 0.61%, 0.55% and 0.43% of the morphemes, respectively, they are not components of other morphemes which they could be mistaken for. In contrast, Morfessor and MorphoChain achieve lower average recalls of 83.3% and 46.3% on the three morphemes, respectively. On another hand, the AG-SS model over-segments words containing *isch* by producing the morpheme *isch* with a 100% recall and a low precision of 39.3%. The reason for that is the high ambiguity of the morpheme, as *isch* is a separate morpheme 37.9% of the time it occurs, in addition to its low relative frequency of 0.67.2%. In contrast, both Morfessor and MorphoChain under-segment the morpheme *isch*, with low recalls of 9.1% and 18.2%, respectively. Moreover, our models tend to over-segment consecutive simple prefixes, e.g., merging *auf+ent* into *aufent*. One explanation is that prefixes are not frequent enough in the data, where the eleven most frequent morphemes are suffixes. While Morfessor shows a similar behavior, MorphoChain tends to split complex prefixes into simple ones.

Georgian The AG-LI model significantly outperforms Morfesssor and MorphoChain in the detection of the top common one-letter morphemes, namely ი, ა, ს, ე, მ, ო and ვ, with an average F1-score of 54.4%, as opposed to 33.1% by Morfessor and 35.7% by MorphoChain. However, these morphemes are highly ambiguous and difficult to be correctly identified by the different models. For instance, all the models achieve a low recall (up to 49.2% by the AG-SS model), when detecting the top two morphemes, ი and ა, where these morphemes appear as part of other bigger morphemes such as ში, და and ვა. In contrast, the morphemes ზე and ად are the top detected morphemes among the most frequent ones. However, the different models achieve relatively low recalls, compared to the performance on other languages, where they tend to under-segment.

Turkish Our models show better detection of one-letter morphemes than Morfessor and MorphoChain. For in-

[7] https://morfessor.readthedocs.io/en/latest/

[8] https://github.com/karthikncode/MorphoChain

Language	BPR				EMMA-2			
	Morfessor	MorphoChain	AG-LI	AG-SS	Morfessor	MorphoChain	AG-LI	AG-SS
English	75.2	69.5	75.5	**80.0**	85.9	82.5	86.3	**88.7**
German	72.8	64.0	78.1	**79.5**	80.9	73.9	84.6	**85.9**
Finnish	62.8	55.7	70.9	**71.1**	73.4	68.9	**77.7**	77.4
Estonian	67.6	61.4	**74.0**	69.0	83.5	75.1	85.3	**85.8**
Georgian	62.1	62.4	**72.7**	72.0	72.1	72.2	78.6	**78.8**
Turkish	64.6	60.6	**78.9**	72.8	61.3	61.1	**69.3**	65.2
Arabic	78.0	77.1	82.5	**90.1**	85.5	85.3	88.4	**93.9**
Zulu	47.5	42.2	65.6	**75.7**	52.5	55.9	69.7	**78.7**
Mexicanero	70.7	69.5	79.4	**82.7**	86.8	86.1	90.1	**92.0**
Nahuatl	58.4	61.5	67.0	**68.3**	80.9	82.9	83.4	**85.2**
Wixarika	70.2	43.1	76.4	**77.9**	72.6	64.0	80.4	**82.5**
Yorem Nokki	63.9	49.3	78.8	**81.1**	81.2	81.1	88.1	**89.1**
Average	66.2	59.7	75.0	**76.7**	76.4	74.1	81.8	**83.6**

Table 4: The results on the **development** sets using the top language-independent (Standard/Cascaded) model (AG-LI) and the top scholar-seeded model (AG-SS) for each language, compared to two baselines; Morfessor and MorphoChain. The results are reported on both the BPR and EMMA-2 F1-scores. The best language-independent result per language and evaluation metric is underlined, while the best overall result per language and evaluation metric is in boldface.

Language	BPR				EMMA-2			
	Morfessor	MorphoChain	AG-LI	AG-SS	Morfessor	MorphoChain	AG-LI	AG-SS
Mexicanero	70.5	64.3	77.7	**78.5**	79.6	77.4	84.6	**87.1**
Nahuatl	61.2	55.9	72.1	**73.7**	73.4	74.8	80.6	**82.8**
Wixarika	72.9	50.3	76.8	**78.2**	71.7	62.3	77.5	**81.2**
Yorem Nokki	71.7	58.0	**81.0**	80.7	79.0	77.6	86.3	**87.0**
Average	69.1	57.1	76.9	**77.8**	75.9	73.0	82.3	**84.5**

Table 5: The results on the **test** sets using the top language-independent (Standard/Cascaded) model (AG-LI) and the top scholar-seeded model (AG-SS) for each language compared to two baselines; Morfessor and MorphoChain. The results are reported on both the BPR and EMMA-2 F1-scores. The best language-independent result per language and evaluation metric is underlined, while the best overall result per language and evaluation metric is in boldface.

stance, our AG-LI model detects the top three one-letter morphemes, *i* and *ı* and *t*, with an average F1-score of 29.9%, as opposed to average F1-scores of 16.1% and 20.1% by Morfessor and MorphoChain, respectively. However, the detection of these morphemes remains a challenge as they are highly ambiguous and usually appear as part of bigger morphemes. On the other hand, despite *ler* (a plural suffix) being a morpheme 82.2% of the time it occurs, our AG-LI model recognizes it successfully only 38.5% of the time, probably because there are other morphemes containing *ler* such as *leri*. Most of such errors occur when *ler* is followed by a vowel. On the other side, MorphoChain has a better detection of the *ler* morpheme, while Morfessor tends to under-segment it. Another example is the morpheme *ma*. Despite the fact that *ma* is the fourth most occurring morpheme in the data, all the models tend to merge it with *sı* due to the frequent occurrence of *ması*. In contrast, the morpheme *yla* is always correctly identified by the AG-LI model due to its low degree of ambiguity (90.0% of the time it is a morpheme), and the fact that it is almost always an ending suffix. On the other hand, the recalls of Morfessor and MorphoChain in detecting *yla* are significantly lower, 27.8% and 55.6%, respectively.

Arabic [9] The precision and recall of our models in the detection of the Arabic morphemes are the highest among the other languages. The AG-LI model detects the one-letter common morphemes efficiently, e.g., the affixes *w*, *t*, *y*, *n*, *p* and *k*, especially they mostly appear in the very beginning or at the end of words. In contrast, both Morfessor and MorphoChain have relatively low recalls and precisions, under 70.0%, in the detection of the verbal prefixes *t*, *y* and *n*. In addition. Morfessor tends to over-segment *w* when it appears in the middle of a word, while MorphoChain over-segments *k* when it is part of a stem. However, our models fail to detect the verbal prefix *s* as it is always followed by another prefix. On another hand, the AG-LI model and MorphoChain tend to over-segment the beginning *m* since many adjectives start with *m*, but it is not an Arabic prefix. It is also noted that some segmentation errors are correct when ignoring the context, while the gold segmentation is based on the context in the PATB corpus. An example is the word *t$bh*, which means either *t+$bh* (she looks like) or *t$bh* (resembling).

[9] We use the Buckwalter Transliteration for better readability (Buckwalter, 2004)

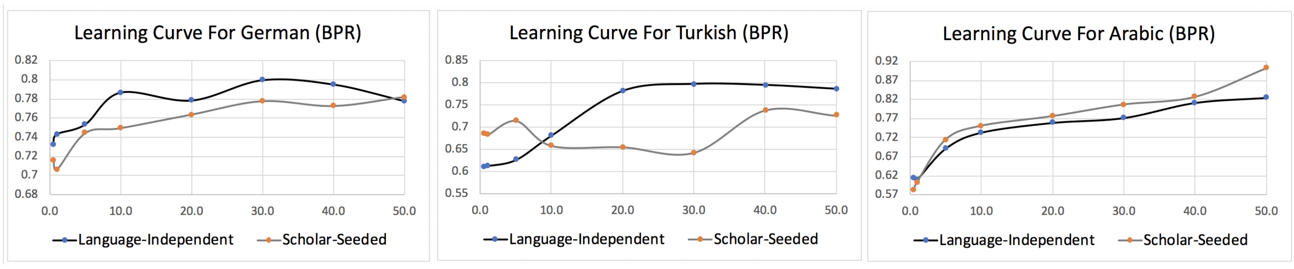


Figure 4: Learning Curves for German, Turkish and Arabic when training on different vocabulary sizes: 500, 1K, 5K, 10K, 20K, 30K, 40K and 50K. The X-axis represents the training size in thousands. The Y-axis represents the BPR F1-scores.

Language	word	Gold Segmentation	AG-LI segmentation	AG-SS Segmentation
German	absonderlicher anfeuchtet tarifarische aufenthalte	ab+sonder+lich+er an+feucht+et tarif+arisch+e auf+ent+halt+e	ab+sonder+lich+er an+feucht+et tarif+arisch+e auf+*enthalt*+e	ab+sonder+lich+er an+feucht+et tarif+*ar*+*isch*+e auf+*enthalt*+e
Georgian	ნაჭამი კურსზე აპირებ თვეშიც	ნა+ჭამ+ი კურს+ზე ა+პირ+ებ თვე+ში+ც	ნა+ჭამ+ი კურს+ზე აპირებ თვე+ში+ც	ნა+ჭამ+ი კურს+ზე ა+პირ+ებ თვე+შიც
Turkish	nedameti pastanelerde oynamasıdır esrarlarıyla	nedamet+i pastane+ler+de oyna+ma+sı+dır esrar+ları+yla	nedamet+i pastane+ler+de oyna+*ması*+dır esrar+ları+yla	nedamet+i pastane+*lerde* oyna+*masıdır* esrar+*larıyla*
Arabic	wqrTAs fAlE$yqp sytmkn mqAtlyn	w+qrTAs f+Al+E$yq+p s+y+tmkn mqAtl+yn	w+qrTAs f+Al+E$yq+p *sy*+tmkn *m*+*qAtl*+yn	w+qrTAs f+Al+*E$yqp* *sy*+tmkn *mqAtlyn*
Zulu	ngamaqembu ngathola ungazi bayofik+a ngisafanisa	nga+ma+qembu ng+a+thol+a u+ng+azi ba+yo+fik+a ngi+sa+fan+is+a	nga+ma+qembu *nga*+*thola* u+*ngazi* ba+yo+*fika* ngi+sa+*fanisa*	nga+ma+qembu *nga*+thol+a u+*nga*+*z*+*i* ba+yo+fik+a ngi+sa+fan+is+a
Wixarika	pütawieya nepexeiya nepexeiya perexeiya	pü+tawie+ya ne+p+e+xeiya ne+p+e+xeiya pe+r+e+xeiya	pü+tawie+ya ne+p+e+xeiya ne+*pe*+xeiya pe+*re*+xeiya	pü+tawie+ya ne+p+e+xeiya ne+*pe*+xeiya pe+*re*+*xei*+*ya*
Yorem Nokki	nechie' usimpo pwertapo tekipanwapeyaka	ne+chi+e' usi+m+po pwerta+po tekipan+wa+peya+ka	ne+chi+e' usi+m+po *pwer*+*tapo* *tekipanwa*+peya+ka	ne+chi+e' usi+m+po pwerta+po *tekipanwa*+peya+ka

Table 6: Examples of correct and *incorrect* segmentation for German, Georgian, Turkish, Arabic, Zulu, Wixarika and Yorem Nokki

Zulu The AG-LI and AG-SS models are able to identify the prefix *nga* 79.6% and 77.6% of the time, respectively, due to the low ambiguity and the high frequency of the morpheme. In contrast, both Morfessor and MorphoChain fail to detect *nga* most of the time, with a low recall of 16.3%. On the other side, our models mostly fail to detect the morpheme *ng* as it is usually part of other morphemes such as *nga* and *ngi*, while MorphoChain tends to over-segment *ng* instead. One observed phenomenon in the case of Zulu is that our models vary widely in their performance. For instance, while the AG-SS model is able to detect the most frequently used morpheme, *a*, correctly 75.8% of the time, the AG-LI model detects it correctly only 14.6% of the time. The AG-LI model's errors occur around *a* often due to under-segmentation, while the AG-SS model is efficient at identifying *a* as a separate morpheme when it is at the end of a word , although it often fails to identify it when it is in the middle. In contrast, MorphoChain can detect the ending *a* morpheme only 6.8% of the time, while Morfessor under-segment it consistently, which highly affects their overall performance. On another hand, all the models fail to detect the *is* morpheme, except the AG-SS model, which is able to detect it only 22.6% of the time. However, all the models show under-segmentation in the case of Zulu.

Wixarika Our AG-LI and AG-SS models detect the two most frequent morphemes, *pü* and *ne*, efficiently with average F1-scores of 86.2% and 94.8%, respectively. In contrast, Morfessor and MorphoChain achieve significantly lower average F1-scores on the two morphemes, 39.6% and 18.5%, in order. However, none of language-independent systems can produce the morphemes *p*, *e* and *r*, which are ranked fifth, sixth and tenth in terms of frequency. In general, the segmentation models tend to under-segment the one-letter morphemes as part of bigger ones when learning from small corpora. On the other side, MorphoChain shows a high degree of under-segmentation, where it achieves zero F1-scores in the detection of seven out of ten most frequent morphemes, and low recalls, up to 27.3%, on the rest of the morphemes.

Yorem Nokki Our AG-LI and AG-SS models are highly efficient at detecting the most frequent morphemes, where the AG-LI model can always detect the morphemes *ne*, *su* and *e'* correctly, while the AG-SS model achieves a 100% recall in the detection of the *k*, *po* and *e'* morphemes. In contrast, Morfessor can only detect the morpheme *e'* correctly in a consistent manner, while MorphoChain achieves a zero F1-score in the detection of five out of the ten most frequent morphemes, namely *k*, *ka*, *ri*, *e'* and *wa*. While the AG-LI and AG-SS models tend to over-segment in Yorem Nokki, MorphoChain under-segments most of the time, which is why it cannot produce many of the short common morphemes. Finally, all the models are inefficient in detecting the morpheme *wa* due to its high degree of ambiguity.

For the analysis of the common segmentation phenomena seen in both Mexicanero and Nahuatl, see Eskander et al. (2019).

5. Related Work

Unsupervised morphological segmentation was first performed by expensive manual rule engineering. An early use of machine learning for morphological segmentation was proposed by Goldsmith (2001) through the use of the Minimum Description Length (MDL) approach. The approach, however, requires some manual work that makes it challenging to generalize across languages.

Morfessor (Creutz and Lagus, 2002), is a commonly used unsupervised and semi-supervised morphological-segmentation framework that utilizes the MDL principal, along with an HMM model, where the morphemes have a hierarchical structure. Another variation of Morfessor is Morfessor FlatCat (Grönroos et al., 2014), which predicts both segmentation and morpheme categories.

Log-linear models have proved successful for the problem of unsupervised morphological segmentation (Poon et al., 2009) with the use of global and contextual features. Another log-linear model is proposed by Narasimhan et al. (2015), where they arrange the words into chains that model the word formation process. A chain starts with a base word and ends with some variant, where predicting the chain of a given word derives its segmentation information.

Johnson et al. (2007) propose Adaptor Grammars, nonparametric Bayesian models that generalize PCFGs. Adaptor Grammars have then become the basis for several unsupervised morphological segmentation systems.

Botha and Blunsom (2013) extend Adaptor Grammars to model non-concatenative morphology, while Sirts and Goldwater (2013) and Eskander et al. (2016) utilize Adaptor Grammars by exploring different grammars and learning settings for language-independent and minimally supervised morphological segmentation. In a follow-up study, Eskander et al. (2018) proposes a machine-learning approach for the automatic identification of the best Adaptor-Grammar learning setups. Eskander et al. (2019) then utilize Adaptor Grammars for the unsupervised morphological segmentation of polysynthetic languages. They show that Adaptor Grammars are highly efficient in low-resource setups.

A comprehensive study that compares different unsupervised and semi-supervised morphological segmentation approaches, including Morfessor, MorphoChain and basic Adaptor-Grammar setups, is conducted by Ruokolainen et al. (2016).

6. Conclusion and Future Work

We presented MorphAGram, a publicly available framework for unsupervised morphological segmentation that is based on Adaptor Grammars. The framework can also benefit from the addition of language-specific information. We conducted an extensive quantitative and qualitative evaluation using two common evaluation metrics, BPR and EMMA-2, on 12 languages that are spread across the typology spectrum (from fusional to polysynthetic and from high-resource to low-resource). We showed that the framework achieves the state-of-the-art results on all languages on both metrics. We also conducted an error analysis to discuss the most common phenomena seen in the segmentation outputs of several languages.

In the future, we plan to conduct an extrinsic evaluation for our morphological-segmentation framework on downstream tasks such as part-of-speech tagging, machine translation, information retrieval and learning cross-lingual embeddings.

7. 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), (contract FA8650-17-C-9117) and the Army Research Laboratory(ARL). The views and conclusions herein are those of the authors and should not be interpreted as necessarily representing official policies, ex-pressed or implied, of ODNI, IARPA, ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copy-right annotation therein.

8. Bibliographical References

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine learning*, 50(1-2):5–43.

Ataman, D., Negri, M., Turchi, M., and Federico, M. (2017). Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English. 108.

Botha, J. A. and Blunsom, P. (2013). Adaptor Grammars for Learning Non- Concatenative Morphology. Association for Computational Linguistics.

Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. ldc cat alog no. Technical report, Ldc2004l02. Technical report.

Comrie, B. (1993). Typology and Reconstruction. *Historical Linguistics: problems and perspectives*, pages 74–97.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2007). Unsupervised Models for Morpheme Segmentation and Morphology Learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, February.

Elsner, M., Charniak, E., and Johnson, M. (2009). Structured Generative Models for Unsupervised Named-Entity Clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172.

Eskander, R., Rambow, O., and Yang, T. (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.

Eskander, R., Rambow, O., and Yang, T. (2018). Extending the Use of Adaptor Grammars for Unsupervised Morphological Segmentation of Unseen Languages. In *Proceedings of the Fifteenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Brussels, Belgium.

Eskander, R., Klavans, J. L., and Muresan, S. (2019). Unsupervised Morphological Segmentation for Low-Resource Polysynthetic Languages. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195.

Goldsmith, J. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational linguistics*, 27(2):153–198.

Grönroos, S.-A., Virpioja, S., Smit, P., and Kurimo, M. (2014). Morfessor FlatCat: An HMM-based Method for Unsupervised and Semi-Supervised Learning of Morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185.

Huang, Y., Zhang, M., and Tan, C. L. (2011). Nonparametric Bayesian mMchine Transliteration with Synchronous Adaptor Grammars. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539.

Ishwaran, H. and James, L. F. (2003). Generalized Weighted Chinese Restaurant Processes for Species Sampling Mixture Models. *Statistica Sinica*, pages 1211–1235.

Johnson, M. and Demuth, K. (2010). Unsupervised phonemic Chinese Word Segmentation using Adaptor Grammars. In *Proceedings of the 23rd international conference on computational linguistics*, pages 528–536. Association for Computational Linguistics.

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007). Adaptor Grammars: a Framework for Specifying Compositional Nonparametric Bayesian Models. In B. Sch"olkopf, et al., editors, *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA. MIT Press.

Johnson, M. (2008a). 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.

Johnson, M. (2008b). 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, June. Association for Computational Linguistics.

Johnson, M. (2008c). Using Adaptor Grammars to Identify Synergies in the Unsupervised Acquisition of Linguistic Structure. In *Proceedings of ACL-08: HLT*, pages 398–406.

Kann, K., Mager Hois, J. M., Meza Ruiz, I. V., and Schütze, H. (2018). Fortification of Neural Morphological Segmentation Models for Polysynthetic Minimal-Resource Languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57.

Klavans, J. L. (2018). *On Clitics and Cliticization: The Interaction of Morphology, Phonology, and Syntax*. Routledge.

Kurimo, M., Virpioja, S., Turunen, V., and Lagus, K. (2010). Morpho Challenge Competition 2005–2010: Evaluations and Results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95. Association for Computational Linguistics.

Maamourio, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Narasimhan, K., Karakos, D., Schwartz, R. M., Tsakalidis, S., and Barzilay, R. (2014). Morphological Segmentation for Keyword Spotting. In *EMNLP*.

Narasimhan, K., Barzilay, R., and Jaakkola, T. (2015). An Unsupervised Method for Uncovering Morphological Chains. In *Twelfth AAAI Conference on Artificial Intelligence*.

Nguyen, T., Vogel, S., and Smith, N. A. (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.

Pitman, J., Yor, M., et al. (1997). The two-parameter Poisson-Dirichlet Distribution Derived from a Stable Subordinator. *The Annals of Probability*, 25(2):855–900.

Pitman, J. (1995). Exchangeable and Partially Exchangeable Random partitions. *Probability theory and related fields*, 102(2):145–158.

Poon, H., Cherry, C., and Toutanova, K. (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, June. Association for Computational Linguistics.

Ruokolainen, T., Kohonen, O., Sirts, K., Grönroos, S.-A., Kurimo, M., and Virpioja, S. (2016). A Comparative Study of Minimally Supervised Morphological Segmentation. *Computational Linguistics*, 42(1):91–120, March.

Sirts, K. and Goldwater, S. (2013). Minimally-Supervised Morphological Segmentation using Adaptor Grammars. *Transactions of the Association for Computational Linguistics*, 1(May):231–242.

Spiegler, S. and Monson, C. (2010). Emma: a Novel Evaluation Metric for Morphological Analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1029–1037. Association for Computational Linguistics.

Spiegler, S., Van Der Spuy, A., and Flach, P. A. (2010). Ukwabelana: An open-source morphological Zulu corpus. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1020–1028. Association for Computational Linguistics.

Virpioja, S., Turunen, V. T., Spiegler, S., Kohonen, O., and Kurimo, M. (2011). Empirical Comparison of Evaluation Methods for Unsupervised Learning of Morphology. *Traitement Automatique des Langues*, 52(2):45–90.

Wong, S.-M. J., Dras, M., and Johnson, M. (2012). Exploring Adaptor Grammars for Native Language Identification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 699–709. Association for Computational Linguistics.