

Unsupervised Morphological Segmentation for Low-Resource Polysynthetic Languages

*Report submitted in fulfillment of the requirements
for the Exploratory Project of*

Second Year B.Tech.

by

Yash Malik

18075065

Under the guidance of

Dr. A.K. Singh



Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2017

Dedicated to

My parents and my teachers.

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.
2. The work has not been submitted for any project.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date: 16th June, 2020

Yash Malik
B.Tech. (Part II) Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled “**Unsupervised Morphological Segmentation for Low-Resource Polysynthetic Languages**” being submitted by **Yash Malik (Roll No. 18075065)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: 20th June, 2020

Dr. A.K. Singh
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

I would like to express my sincere gratitude to my supervising Professor Dr. A.K. Singh and guiding mentor Mr. Rajesh Kumar Mundotiya for their constant guidance and support during the whole project work.

Place: IIT (BHU) Varanasi

Date: 16th June, 2020

Yash Malik

Abstract

Polysynthetic Languages pose a challenge for morphological analysis due to their very high morpheme-to-word ratio and very high inflection in word formation. These languages are morphologically rich and are highly synthetic, i.e., single words can be composed of many morphemes. In extreme cases the entire sentence consist of only one single token. Morphological segmentation in such cases becomes a very challenging task when we consider the fact that the training data can be extremely scarce for such languages.

This report presents some unsupervised approaches for morphological segmentation of low-resource polysynthetic languages based on Adaptor Grammars(AG) framework [1], as proposed by Eskander et al. (2019) in their research paper [2].

Contents

List of Figures	x
List of Tables	xi
1 Introduction	2
1.1 Overview	2
1.2 Motivation of the Research Work	3
1.3 Organisation of the Report	3
2 Prerequisite Knowledge	4
2.1 Basic terms and their definitions	4
2.2 Adaptor Grammars	5
3 Project Work	7
3.1 Problem Definition	7
3.2 Languages and Datasets	8
3.2.1 Language Typology and Morphological Analysis	8
3.2.2 Morphological Segmentation Datasets	10
3.3 Describing the Solution Setup	11
3.3.1 Using Adaptor Grammars for Polysynthetic Languages	12
3.4 Defining Grammars	12
3.5 Training The Model	14

CONTENTS

3.5.1	Inputs	14
3.5.2	Learning Settings	15
3.6	Evaluation and Results	18
3.6.1	Evaluation Setups	18
3.6.2	Evaluation Metrics	18
3.6.3	Baselines	19
3.6.4	System Performance	20
4	Conclusions and Discussion	21
	Bibliography	25

List of Figures

3.1	The representation of English word <i>irreplaceables</i> segmented using <i>PrStSu+SM</i> , from Eskander et al. (2020) [3]	13
3.2	The representation of English word <i>irreplaceables</i> segmented using <i>PrStSu2a+SM</i> , from Eskander et al. (2020) [3]	13
3.3	The representation of English word <i>irreplaceables</i> segmented using <i>Morph+SM</i> , from Eskander et al. (2020) [3]	14
3.4	The standard <i>PrStSu+SM</i> grammar (note the simple recursive rules of the grammar and the relation with fig 3.1)	15
3.5	The scholar-seeded <i>PrStSu+SM</i> grammar for English (note the addition of flagged lines and knowledge seeds in the end) (the flags are not a part of the grammar)	17
4.1	BPR scores of each language (from table 3.4)	21
4.2	EMMA-2 scores of each language (from table 3.4)	22
4.3	Comparison of BPR F1-scores of the setups, across all languages (from table 3.5)	22
4.4	Comparison of EMMA-2 F1-scores of the setups, across all languages (from table 3.5)	23
4.5	Comparison of the average F1 score of the setups (BPR scores on the left side and EMMA-2 scores on the right side of the graph)	23

LIST OF FIGURES

List of Tables

3.1	Number of words in train, dev, test splits from Kann et al. (2018) datasets	10
3.2	Number of words in train and test splits in the prepared datasets (for transductive learning)	10
3.3	Result on Test sets using the AG based LIMS model (AG-LIMS). . .	20
3.4	Result on Test sets using the AG based LIMS model (AG-LIMS) for each language compared to two baselines; Morfessor and MorphoChain. The results are reported on both the BPR and EMMA-2 F1-scores. .	20

Chapter 1

Introduction

1.1 Overview

In Natural Language Processing (NLP) morphological segmentation is the task of segmenting words into meaning-bearing morphemes. A morpheme is the smallest unit of language that has meaning or function, which includes words, prefixes, affixes, and other word structures that impart meaning. Morphological segmentation is an important task NLP and has many direct or indirect applications in various fields such as text classification, machine translation, speech recognition, etc.

“Polysynthetic languages pose unique challenges for computational approaches, including machine translation and morphological analysis, due to the root-morpheme complexity and to word class gradations.”[2] Some of the previous approaches to this problem are based on finite state transducers, and supervised machine learning approaches, particularly deep learning approaches.

This report presents an unsupervised approach for morphological segmentation of polysynthetic languages based on Adaptor Grammars framework [1]. We show that the Adaptor Grammars based approaches outperform other unsupervised methods — Morfessor [4] and MorphoChain [5] — which are considered as the state-of-the-art baselines.

1.2 Motivation of the Research Work

Computational morphological segmentation has been an active research topic as it forms an important basis for many natural language processing tasks.

Due to the high cost of manually labeling data for morphology and the increasing interest in low-resource and/or endangered languages, where they lack adequate morphologically annotated resources; need arises to find an unsupervised approach for the task of morphological segmentation, especially for low-resource languages. The main goal of this report is to examine the success of Adaptor Grammars for unsupervised morphological segmentation when applied to polysynthetic languages, where the morphology is synthetically complex (not simply agglutinative), and where resources are minimal.

1.3 Organisation of the Report

The first chapter contains some important definitions and description of terms and tools used and associated with the field of natural language processing and morphological segmentation; quoted from relevant sources.

The next chapter describes the project work. The first section describes the problem and challenges associated with it. In the following sections we describe the proposed solution setup followed by sections detailing the various aspects of the setup. The last section of this chapter describes evaluation environment and the metrics used and finally reports the results of our setup versus the state-of-the-art baselines.

We wrap up by discussing and concluding our results in the last chapter. We also include the possibility of extension of this setup for Indian languages in future.

Chapter 2

Prerequisite Knowledge

In this chapter we define and explain some of the basic terminologies and tools that we continuously use in the report. All the definitions and explanations are quoted from relevant sources.

2.1 Basic terms and their definitions

- **Morpheme** : “A morpheme is the smallest meaningful unit in a language.”[6]
- **Affix** : “In linguistics, an affix is a morpheme that is attached to a word stem to form a new word or word form.”[7]
- **Inflection** : “In linguistic morphology, inflection (or inflexion) is a process of word formation, in which a word is modified to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, mood, animacy, and definiteness.”[8]
- **Agglutination** : “Agglutination is a linguistic process pertaining to derivational morphology in which complex words are formed by stringing together morphemes without changing them in spelling or phonetics.”[9] “An agglutinating language (e.g., Turkish or Finnish) is one in which word forms can be easily

2.2. Adaptor Grammars

and clearly segmented into individual morphs, each of which represents a single grammatical category.”[3]

- **Isolating Language** : “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.”[3]
- **Synthetic Language** : “A synthetic language uses inflection or agglutination to express syntactic relationships within a sentence. Inflection is the addition of morphemes to a root word that assigns grammatical property to that word, while agglutination is the combination of two or more morphemes into one word.”[10]
- **Polysynthetic Language** : “Polysynthetic languages are morphologically rich languages which are highly synthetic, i.e., single words can be composed of many individual morphemes.”[11]
- **Context Free Grammar (CFG)** : “In formal language theory, a context-free grammar (CFG) is a formal grammar in which every production rule is of the form

$$A \rightarrow \alpha$$

where A is a single nonterminal symbol, and α is a string of terminals and/or nonterminals (α can be empty). A formal grammar is considered ”context free” when its production rules can be applied regardless of the context of a nonterminal.”[12]

2.2 Adaptor Grammars

“Adaptor Grammars are a framework for specifying nonparametric Bayesian models that can be used to learn latent tree structures from a corpus of strings. There are two

components to an AG model: the *base distribution*, which is just a PCFG, and the *adaptor*, which “adapts” the probabilities assigned to individual subtrees under the PCFG model, such that the probability of a subtree under the complete model may be considerably higher than the product of the probabilities of the PCFG rules required to construct it. Although in principle the adaptor can be any function that maps one distribution onto another, Johnson et al. (2007) use a Pitman-Yor Process (PYP) (Pitman and Yor, 1997) as the adaptor because it acts as a caching model. Under a PYP AG model, the posterior probability of a particular subtree will be roughly proportional to the number of times that subtree occurs in the current analysis of the data (with the probability of unseen subtrees being computed under the base PCFG distribution).

An AG model can be defined by specifying the CFG rules (the support for the base distribution) and indicating which non-terminals are “adapted”, i.e., can serve as the root of a cached subtree. Given this definition and an input corpus of strings, Markov chain Monte Carlo samplers can be used to infer the posterior distribution over trees (and all hyperparameters of the model, including PCFG probabilities in the base distribution and PYP hyperparameters). Any frequently recurring substring (e.g., a common prefix) will tend to be parsed consistently, as this permits the model to treat the subtree spanning that string as a cached subtree, assigning it higher probability than under the PCFG distribution.”[13]

Chapter 3

Project Work

3.1 Problem Definition

Design a statistical machine learning algorithm to split a given word into the surface forms of its smallest meaning-bearing units, morphemes.

Examples (English Language):

treatable = *treat* + *able*

carelessness = *care* + *less* + *ness*

irresponsible = *ir* + *respons* + *ible*

unacceptability = *un* + *accept* + *abil* + *ity*

For this particular task, we need to segment words from the four Uto-Aztecan languages: *Mexicanero* (**MX**), *Nahuatl* (**NH**), *Wixarika* (**WX**) and *Yorem Nokki* (**YN**) [11].

We also test various (unsupervised) learning settings for Adaptor Grammars based learning and try to estimate the effect of availability of external linguistic knowledge with our unsegmented training data (semi-supervised).

3.2 Languages and Datasets

This section describes the relation between language typology and morphological analysis followed by description of datasets and method of preparation of datasets.

3.2.1 Language Typology and Morphological Analysis

In linguistic typology, the languages range from: isolating/analytic to synthetic to polysynthetic. Agglutinating refers to the clarity of boundaries between morphemes.

“The more specific gradation is: *agglutinating* to *mildly fusional* to *fusional*. Thus a language might be characterized overall as *polysynthetic and agglutinating*, i.e. generally a high number of morphemes per word, with clear boundaries between morphemes and thus easily segmentable. Another language might be characterized as *polysynthetic and fusional*, so again, many morphemes per word, but many phonological and other processes so it is difficult to segment morphemes.”[2]

Description of the four Yuto-Aztecan languages (which are considered for analysis):

- “**Mexicanero** is a Western Peripheral Nahuatl variant, spoken in the Mexican state of Durango by *approximately one thousand people*.”[11]
- “**Nahuatl** is a large subgroup of the Yuto-Aztecan language family, and, including all of its variants, the most spoken native language in Mexico. The data collected for this work belongs to the Oriental branch spoken by *70 thousand people* in Northern Puebla.”[11]
- “**Wixarika** is a language spoken in the states of Jalisco, Nayarit, Durango and Zacatecas in Central West Mexico by approximately *fifty thousand people*.”[11]
- “**Yorem Nokki** is part of Taracachita subgroup of the Yuto-Aztecan language family. Its Southern dialect is spoken by close to *forty thousand people* in the

3.2. Languages and Datasets

Mexican states of Sinaloa and Sonora, while its Northern dialect has about twenty thousand speakers. In this work, we consider the Southern dialect.”[11]

All these four languages are classified as *polysynthetic* languages.[3]

“Typically, polysynthetic languages demonstrate **holophrasis**, i.e. the ability of *an entire sentence* to be expressed as what is considered by native speakers to be *just one word*” [2]

Consider as an example a word from Westen Greenlandic, an exclusively suffixing polysynthetic language (from (“Polysynthetic language. ”, wikipedia.org)) [14]:

- *Aliikkusersuillammassuaanerartassagaluarpaalli*

aliikku - sersu - i - lllamas - sua - a - nerar - ta - ssa - galuar - paal - li

entertainment - provide - one.good.at - say.that - sure.but - but

‘However, they will say that he is a great entertainer, but ...’

Consider the following example from *WX*, one of the languages in the dataset for this report, as an illustration to show this complexity (from ((Mager et al., 2018c)) [15]:

(Note: ‘+’ is a vowel in Wixarika language)

- *yuhutame nep+weiwa*

yu - huta - me ne - p+ - we - iwa

an - two - ns 1sg:s - asi - 2pl:o - brother

I have two brothers.

The morphological analysis of polysynthetic languages thus is challenging due to the root-morpheme complexity. This property of polysynthetic languages makes the task of surface segmentation complex but also relevant for further analysis.

3.2.2 Morphological Segmentation Datasets

Morphological data for polysynthetic languages are scarce. Kann et al. (2018) have made a step by releasing a small set of morphologically segmented datasets, although even in these carefully curated datasets, the distinction between affix and clitic is not always indicated. [11] “These datasets were constructed so they include both segmentable as well as non-segmentable words to ensure that methods can correctly decide against splitting up single morphemes.” [2]

As we are using the data in an unsupervised way, we only use the unsegmented data to prepare the training set. There are two different possible approaches for the learning, 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 ”. [3] **We use the transductive mode of learning in our experiments**, therefore we include the unsegmented words from the test sets in our training sets.

	Mexicanero	Nahuatl	Wixarika	Yorem N.
train	427	540	665	511
dev	106	134	176	127
test	355	449	553	425

Table 3.1: Number of words in train, dev, test splits from Kann et al. (2018) datasets

	Mexicanero	Nahuatl	Wixarika	Yorem N.
Train set	888	1123	1394	1063
Test set	461	583	729	552

Table 3.2: Number of words in train and test splits in the prepared datasets (for transductive learning)

We do not use the segmented version of the data for training (our approach is unsupervised). The datasets provided by Kann et al. (2018) [11] consist of the train

3.3. Describing the Solution Setup

set, development set, and test set. Each source file contains one word per line, and the corresponding target file contains their segmentations. We use these datasets to prepare the datasets for our experiment. We use the unsegmented words (from all the three sets) to prepare our Training set, and we use the words as well as their respective segmentations to prepare our Test set (from dev and train sets). Table 3.1 and 3.2 describe this information and the respective word counts for each language.

Note: The datasets from Kann et al. have been slightly processed for clarity purposes and to satisfy the criteria for the input-output format of our application and of the evaluation software. Notably, we use a space character ' ' instead of a '!' to separate the segments, and space between the letters of a word is removed. e.g.,

$$p ! u ! k a ! w e \implies p u k a w e$$

Both the datasets (original and processed) are provided along with this report.

3.3 Describing the Solution Setup

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. AGs provide a simple framework to implement Bayesian nonparametric learning of grammars and are usually trained in an unsupervised manner using sampling techniques. (Eskander et al., 2020)”

We try several AG learning setups:

1. use the standard grammar in completely unsupervised sense

2. approximate the effect of having some linguistic knowledge
3. use the *best on average* AG setup ‘**LIMS**’ from Eskander et al. (2016) [16]

3.3.1 Using Adaptor Grammars for Polysynthetic Languages

“An Adaptor Grammar is typically composed of a PCFG and an adaptor that adapts the probabilities of individual subtrees. For morphological segmentation, a PCFG is a morphological grammar that specifies word structure, where AGs learn latent tree structures given a list of words.” [2]

The AGs take as input the vocabulary of the language we want to learn and the grammar rules the sampler has to follow.

- The vocabulary input is a list of unsegmented words.
- The grammar input consists of rules along with the adaptation information (see section 3.4 for details).

We briefly outline the experiment setup below (and describe them in detail in their respective sections):

Grammars We use the nine grammars from Eskander et al. (2016) [16].

Learning Settings We consider the three learning settings in (Eskander et al., 2016) [16]: *Standard*, *Scholar-seeded Knowledge* and *Cascaded*.

3.4 Defining Grammars

The first step in learning morphological segmentation using Adaptor Grammars is to define the grammar. We use the nine grammars described by Eskander et al. (2016)[16].

3.4. Defining Grammars

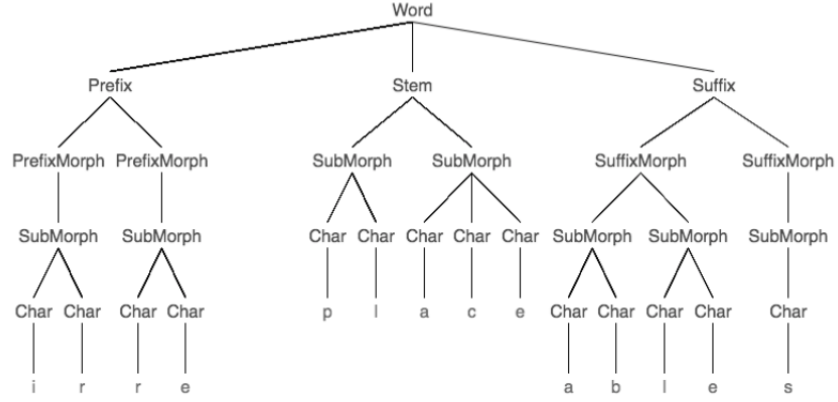


Figure 3.1: The representation of English word *irreplaceables* segmented using *PrStSu+SM*, from Eskander et al. (2020) [3]

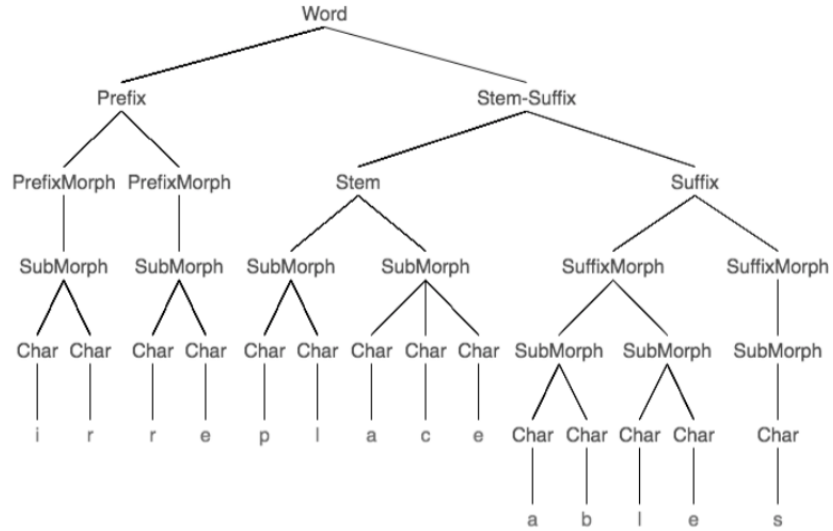


Figure 3.2: The representation of English word *irreplaceables* segmented using *PrStSu2a+SM*, from Eskander et al. (2020) [3]

Figures 3.1, 3.2 and 3.3 show the grammar trees of three of the nine grammars used: *PrStSu+SM*, *PrStSu2a+SM* and *Morph+SM*, representing the English word *irreplaceables*. (*Pr*, *St* and *Su* refer to *Prefix*, *Stem* and *Suffix*, respectively. *SM* represents *sub-morph*)

For more details about the specifications of these grammars and other grammars, see Eskander et al. (2016) [16]

“In addition to defining the main grammar, each production rule has to be as-

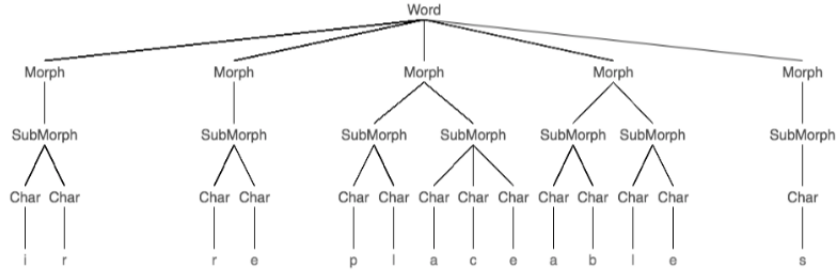


Figure 3.3: The representation of English word *irreplaceables* segmented using *Morph+SM*, from Eskander et al. (2020) [3]

sociated 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 [17]. 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 [18] 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.”[3]

For more details, see Johnson et al. (2007) [1].

3.5 Training The Model

3.5.1 Inputs

The main inputs to the learner are the grammar and the vocabulary of the language we want to learn the segmentation for.

It is to be noted that the standard grammar is a comprehensive set of rules and is entirely independent of the language. Also, note that as **we are using the transductive learning method**, the test set’s unsegmented words should be included

3.5. Training The Model

in the train set.

We use $\wedge \wedge \wedge$ and $\$ \$ \$$ to indicate the beginning and end of words respectively.

```
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 3.4: The standard *PrStSu+SM* grammar (note the simple recursive rules of the grammar and the relation with fig 3.1)

The input grammar to the AGs should follow the format similar to one in Fig. 3.4.

3.5.2 Learning Settings

Escander et al., (2016) [16] define three learning settings: *Standard*, *Scholar-seeded* and *Cascaded*. We briefly define them here.

- **Standard:** The standard setting is language independent. The grammar does not have any language-specific rules, and *the learning is fully unsupervised*. Fig.

3.4 shows the input of *PrStSu+SM* grammar in the standard mode.

- **Scholar-seeded:** When 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 [3]. Fig. 3.5 shows the input of *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 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.” [3]

Note: “A production rule that is not preceded by parameters in Fig. 3.4 and Fig. 3.5 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.” [3]


Also, note that the grammars in Fig. 3.4, and Fig. 3.5 are not ready to use as input for the Adaptor Grammars framework. The *Char* (in grammar rules) needs to be assigned to every letter in the alphabet of the language for which we want to learn the segmentation. However, this process can be automated by picking up the characters from the training set and defining the grammar then.

Note: The nine grammars used in this experiment are provided for both *Standard*


3.5. Training The Model

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

Prefix --> ^^^
Prefix --> ^^^ PrefixMorphs

1 1 PrefixMorphs --> PrefixMorph PrefixMorphs
1 1 PrefixMorphs --> PrefixMorph
1 1 PrefixMorph --> SeededPrefixMorph 
PrefixMorph --> SubMorphs

Stem --> SubMorphs

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

1 1 SubMorphs --> SubMorph SubMorphs
1 1 SubMorphs --> SubMorph
SubMorph --> Chars

1 1 Chars --> Char
1 1 Chars --> Char Chars

1 1 SeededPrefixMorph --> a n t i
1 1 SeededPrefixMorph --> s e m i
1 1 SeededPrefixMorph --> p r e
1 1 SeededPrefixMorph --> d i s
1 1 SeededPrefixMorph --> n o n

1 1 SeededSuffixMorph --> e d
1 1 SeededSuffixMorph --> n e s s
1 1 SeededSuffixMorph --> m e n t
1 1 SeededSuffixMorph --> i n g
1 1 SeededSuffixMorph --> s
1 1 SeededSuffixMorph --> ' s
1 1 SeededSuffixMorph --> t i o n
```

Figure 3.5: The scholar-seeded *PrStSu+SM* grammar for English (note the addition of flagged lines and knowledge seeds in the end) (the flags are not a part of the grammar)

and *Scholar-seeded* settings along with the report. The *Cascaded* learning setting also works using these two grammars.

3.6 Evaluation and Results

In this section we evaluate our morphological segmentation setup, qualitatively and analytically. We process the datasets outlined in section 3.2 . We first describe the evaluation setup, then we describe the evaluation metrics and the baselines. We conclude by comparing our results to the state-of-the-art baselines.

3.6.1 Evaluation Setups

- We conduct the evaluation for LIMS setup, as proposed by Eskander et al., (2016) [16]
- We conduct the evaluation in a transductive learning scenario, where the un-segmented test words are included in our training set.
- We run the sampler for 500 iterations for all languages (the performance is almost similar to that at 1000 iterations).
- No annealing is used as it does not improve the results. All the parameters to the Adaptor Grammars are automatically inferred (we do not specify any default value for any of the hyperparameters).
- We compute the results as the average of five runs.

3.6.2 Evaluation Metrics

We evaluate the performance of our setup using two metrics: Boundary Precision and Recall (**BPR**) and Evaluation Metric for Morphological Analysis - 2 (**EMMA-2**)[19].

“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.” [3]

3.6.3 Baselines

We evaluate our system versus two **state-of-the-art** baselines: **Morfessor** [4] and **MorphoChain** [5].

- Morfessor is a commonly used framework for unsupervised and semi-supervised morphological segmentation and is publicly available for free¹.
- MorphoChain is another publicly available system for unsupervised morphological segmentation².

1: (<https://morfessor.readthedocs.io/en/latest/>)

2: (<https://github.com/karthikncode/MorphoChain>)

Note: All the data for the results of Baselines has been taken from Eskander et al., (2020) [3].

3.6.4 System Performance

Table 3.3 reports the performance of our system and Table 3.4 reports our performance compared to Morfessor and MorphoChain based on the respective $F1$ - scores for the four polysynthetic languages when tested on Test sets. The results by our system are under the **AG-LIMS** column.

Fig. 4.1 and 4.2 show the graphical representations of Table 3.3, and Fig. 4.3 and 4.4 show the graphical representations of Table 3.4. (Note: For our system, we only evaluate the results for the LIMS setup as proposed by Eskander et al., (2016). [16])

Language	BPR			EMMA-2		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Mexicanero	67.8	77.3	71.8	85.6	81.8	82.4
Nahuatl	65.1	72.3	67.4	81.1	78.0	78.6
Wixarika	84.3	66.8	73.3	87.2	65.5	72.1
Yorem Nokki	84.0	75.9	78.5	93.1	80.8	84.8

Table 3.3: Result on Test sets using the AG based LIMS model (AG-LIMS).

Language	BPR			EMMA-2		
	Morfessor	MorphoChain	AG-LIMS	Morfessor	MorphoChain	AG-LIMS
Mexicanero	70.5	64.3	71.8	79.6	77.4	82.4
Nahuatl	61.2	55.9	67.4	73.4	74.8	78.6
Wixarika	72.9	50.3	73.3	71.7	62.3	72.1
Yorem Nokki	71.7	58.0	78.5	79.0	77.6	84.8
Average	69.1	57.1	72.8	75.9	73.0	79.5

Table 3.4: Result on Test sets using the AG based LIMS model (AG-LIMS) for each language compared to two baselines; Morfessor and MorphoChain. The results are reported on both the BPR and EMMA-2 $F1$ -scores.

Chapter 4

Conclusions and Discussion

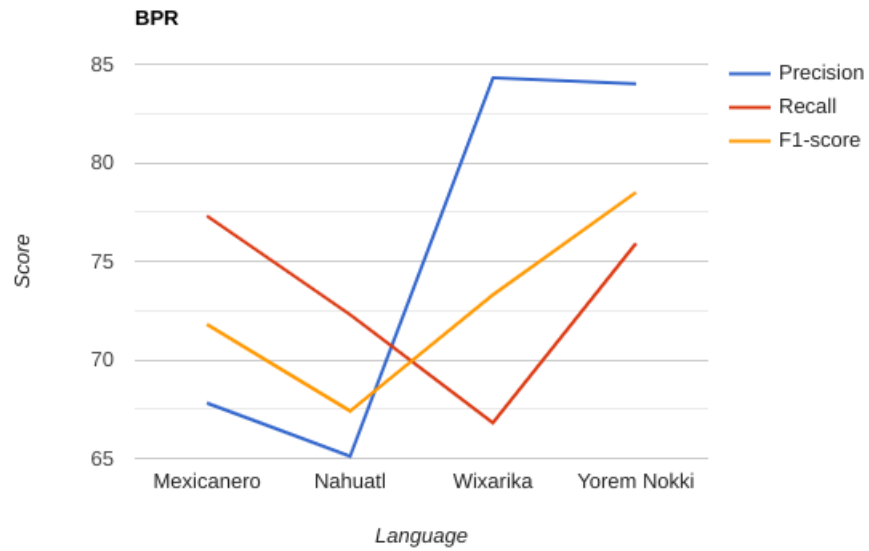


Figure 4.1: BPR scores of each language (from table 3.4)

As can be seen from Table 3.4, we constantly obtain better results versus Morfessor and MorphoChain, for each language when tested using BPR and EMMA-2 metrics. The AG-LIMS setup achieves the best result on all languages in unsupervised manner. The AG-LIMS setup achieves absolute average F1-score increases of 3.7% and 15.7% over Morfessor and MorphoChain respectively, when using BPR metric.

We also establish that Adaptor Grammars based system is able to **generalize and learn well from a small amount of data**, which is the case when learning the segmentation for the polysynthetic languages.

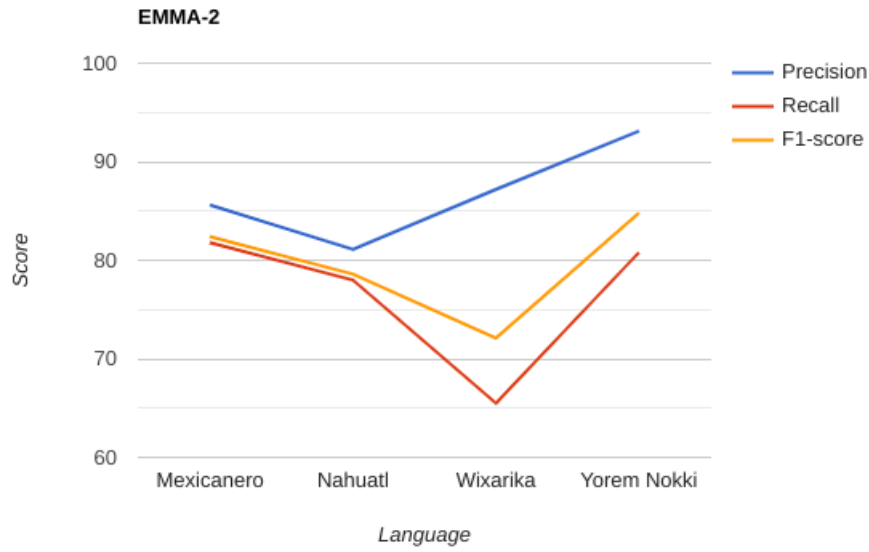


Figure 4.2: EMMA-2 scores of each language (from table 3.4)

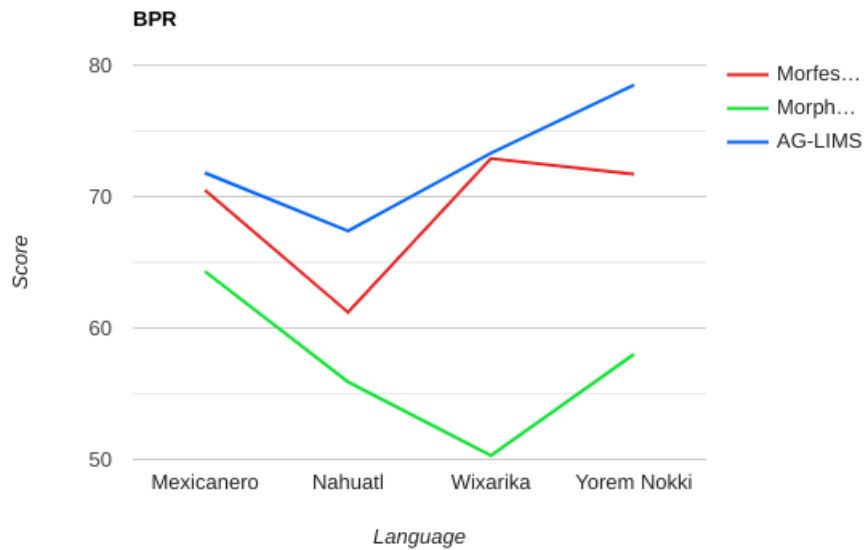


Figure 4.3: Comparison of BPR F1-scores of the setups, across all languages (from table 3.5)

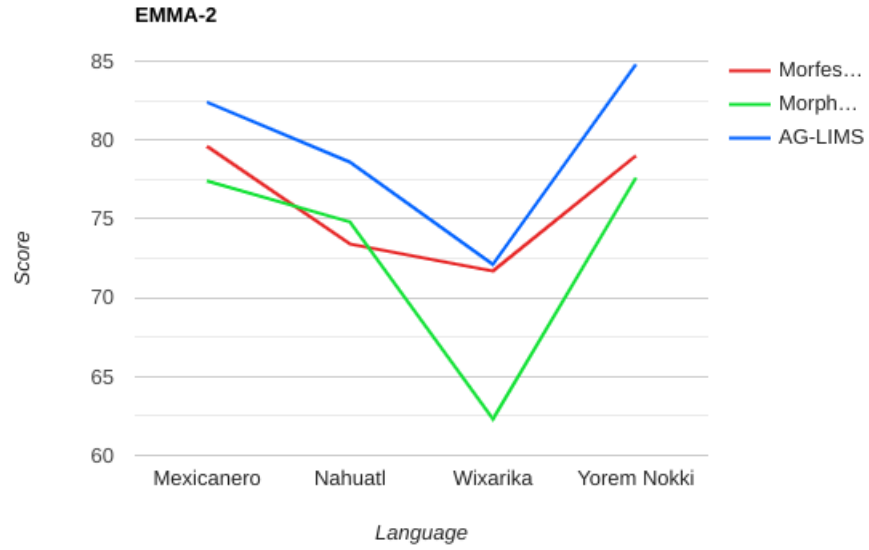


Figure 4.4: Comparison of EMMA-2 F1-scores of the setups, across all languages (from table 3.5)

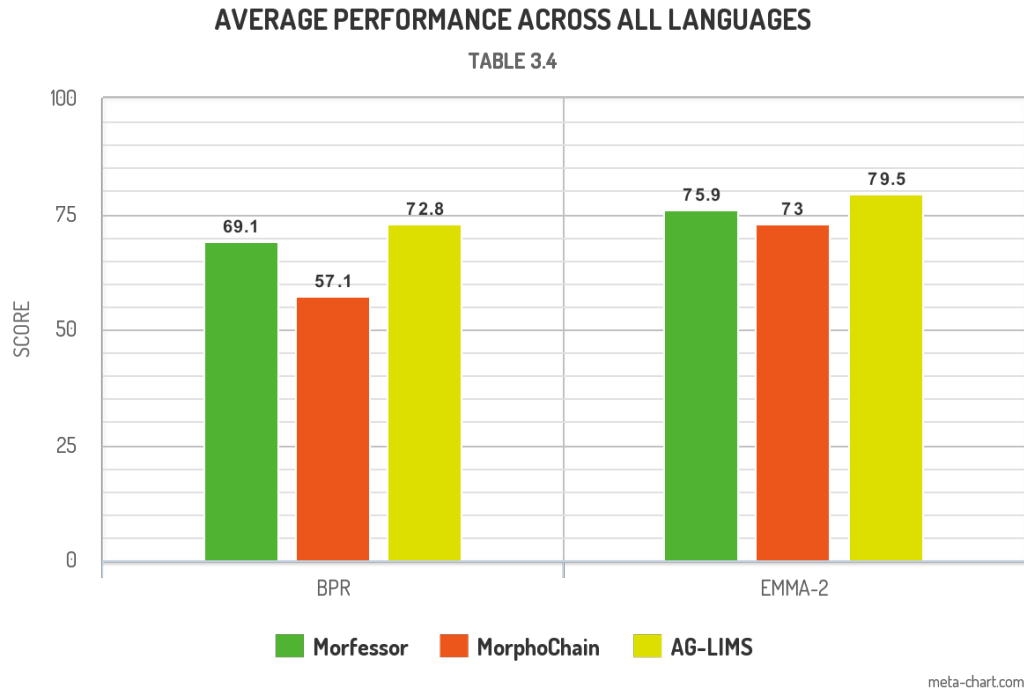


Figure 4.5: Comparison of the average F1 score of the setups (BPR scores on the left side and EMMA-2 scores on the right side of the graph)

Result Analysis and scope for Future Work

Unsupervised approaches based on Adaptor Grammars show promise in the area of morphological segmentation of low-resource polysynthetic languages.

- We showed that the setup achieves the state-of-the-art results on all languages on both metrics.
- We showed that even with very small amount of unsegmented data, the setup is able to produce promising results.

In future this setup can be used for morphological segmentation of some Indian agglutinative languages (for e.g., Dravidian languages). Large morphological dataset is not available for most Indian languages, hence the described setup is promising for this task.

Bibliography

- [1] M. Johnson, T. L. Griffiths, and S. Goldwater, “Adaptor grammars: A framework for specifying compositional nonparametric bayesian models,” in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 641–648. [Online]. Available: <http://papers.nips.cc/paper/3101-adaptor-grammars-a-framework-for-specifying-compositional-nonparametric-bayesian-models.pdf>
- [2] R. Eskander, J. Klavans, and S. Muresan, “Unsupervised morphological segmentation for low-resource polysynthetic languages,” in *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 189–195. [Online]. Available: <https://www.aclweb.org/anthology/W19-4222>
- [3] R. Eskander, F. Callejas, E. Nichols, J. Klavans, and S. Muresan, “MorphAGram, evaluation and framework for unsupervised morphological segmentation,” in *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 7112–7122. [Online]. Available: <https://www.aclweb.org/anthology/2020.lrec-1.879>
- [4] M. Creutz and K. Lagus, “Unsupervised models for morpheme segmentation and morphology learning,” *ACM Trans. Speech Lang. Process.*, vol. 4, pp. 3:1–3:34, 02 2007.
- [5] K. Narasimhan, R. Barzilay, and T. S. Jaakkola, “An unsupervised method for uncovering morphological chains,” *CoRR*, vol. abs/1503.02335, 2015. [Online]. Available: <http://arxiv.org/abs/1503.02335>
- [6] Wikipedia contributors, “Morpheme — Wikipedia, the free encyclopedia,” <https://en.wikipedia.org/w/index.php?title=Morpheme&oldid=960634482>, 2020, [Online; accessed 19-June-2020].
- [7] —, “Affix — Wikipedia, the free encyclopedia,” <https://en.wikipedia.org/w/index.php?title=Affix&oldid=961671292>, 2020, [Online; accessed 18-June-2020].

-
- [8] —, “Inflection — Wikipedia, the free encyclopedia,” <https://en.wikipedia.org/w/index.php?title=Inflection&oldid=955388236>, 2020, [Online; accessed 20-June-2020].
 - [9] —, “Agglutination — Wikipedia, the free encyclopedia,” <https://en.wikipedia.org/w/index.php?title=Agglutination&oldid=960909785>, 2020, [Online; accessed 20-June-2020].
 - [10] —, “Synthetic language — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=Synthetic_language&oldid=961702857, 2020, [Online; accessed 20-June-2020].
 - [11] K. Kann, J. M. Mager Hois, I. V. Meza-Ruiz, and H. Schütze, “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)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 47–57. [Online]. Available: <https://www.aclweb.org/anthology/N18-1005>
 - [12] Wikipedia contributors, “Context-free grammar — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=Context-free_grammar&oldid=959884744, 2020, [Online; accessed 20-June-2020].
 - [13] K. Sirts and S. Goldwater, “Minimally-supervised morphological segmentation using adaptor grammars,” *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 255–266, 2013. [Online]. Available: <https://www.aclweb.org/anthology/Q13-1021>
 - [14] Wikipedia contributors, “Polysynthetic language — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=Polysynthetic_language&oldid=955997075, 2020, [Online; accessed 18-June-2020].
 - [15] M. Mager, X. Gutierrez-Vasques, G. Sierra, and I. Meza-Ruiz, “Challenges of language technologies for the indigenous languages of the Americas,” in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 55–69. [Online]. Available: <https://www.aclweb.org/anthology/C18-1006>
 - [16] R. Eskander, O. Rambow, and T. Yang, “Extending the use of adaptor grammars for unsupervised morphological segmentation of unseen languages,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 900–910. [Online]. Available: <https://www.aclweb.org/anthology/C16-1086>

- [17] J. Pitman and M. Yor, “The two-parameter poisson-dirichlet distribution derived from a stable subordinator,” *Ann. Probab.*, vol. 25, no. 2, pp. 855–900, 04 1997. [Online]. Available: <https://doi.org/10.1214/aop/1024404422>
- [18] H. Ishwaran and L. F. James, “Generalized weighted chinese restaurant processes for species sampling mixture models,” *STATISTICA SINICA*, vol. 13, p. 2003, 2003.
- [19] S. Virpioja, V. T. Turunen, S. Spiegler, O. Kohonen, and M. Kurimo, “Empirical comparison of evaluation methods for unsupervised learning of morphology,” *Traitement Automatique des Langues*, vol. 52, no. 2, pp. 45–90, 2011.