Unsupervised Morphological Segmentation and Part-of-Speech Tagging for Low-Resource Scenarios

Ramy Eskander

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Unsupervised Morphological Segmentation and Part-of-Speech Tagging for Low-Resource Scenarios

Ramy Eskander

With the high cost of manually labeling data and the increasing interest in low-resource languages, for which human annotators might not be even available, unsupervised approaches have become essential for processing a typologically diverse set of languages, whether high-resource or low-resource. In this work, we propose new fully unsupervised approaches for two tasks in morphology: unsupervised morphological segmentation and unsupervised cross-lingual part-of-speech (POS) tagging, which have been two essential subtasks for several downstream NLP applications, such as machine translation, speech recognition, information extraction and question answering.

We propose a new unsupervised morphological-segmentation approach that utilizes Adaptor Grammars (AGs), nonparametric Bayesian models that generalize probabilistic context-free grammars (PCFGs), where a PCFG models word structure in the task of morphological segmentation. We implement the approach as a publicly available morphological-segmentation framework, *MorphAGram*, that enables unsupervised morphological segmentation through the use of several proposed language-independent grammars. In addition, the framework allows for the use of scholar knowledge, when available, in the form of affixes that can be seeded into the grammars. The framework handles the cases when the scholar-seeded knowledge is either generated from language resources, possibly by someone who does not know the language, as weak linguistic

priors, or generated by an expert in the underlying language as strong linguistic priors. Another form of linguistic priors is the design of a grammar that models language-dependent specifications. We also propose a fully unsupervised learning setting that approximates the effect of scholar-seeded knowledge through self-training. Moreover, since there is no single grammar that works best across all languages, we propose an approach that picks a nearly optimal configuration (a learning setting and a grammar) for an unseen language, a language that is not part of the development. Finally, we examine multilingual learning for unsupervised morphological segmentation in low-resource setups.

For unsupervised POS tagging, two cross-lingual approaches have been widely adapted: 1) annotation projection, where POS annotations are projected across an aligned parallel text from a source language for which a POS tagger is accessible to the target one prior to training a POS model; and 2) zero-shot model transfer, where a model of a source language is directly applied on texts in the target language. We propose an end-to-end architecture for unsupervised cross-lingual POS tagging via annotation projection in truly low-resource scenarios that do not assume access to parallel corpora that are large in size or represent a specific domain. We integrate and expand the best practices in alignment and projection and design a rich neural architecture that exploits non-contextualized and transformer-based contextualized word embeddings, affix embeddings and word-cluster embeddings. Additionally, since parallel data might be available between the target language and multiple source ones, as in the case of the Bible, we propose different approaches for learning from multiple sources. Finally, we combine our work on unsupervised morphological segmentation and unsupervised cross-lingual POS tagging by conducting unsupervised stem-based cross-lingual POS tagging via annotation projection, which relies on the stem as the core unit of abstraction for alignment and projection, which is beneficial to low-resource morphologically complex languages. We also examine morpheme-based alignment and projection, the use of linguistic priors towards better POS models and the use of segmentation information as learning features in the neural architecture.

We conduct comprehensive evaluation and analysis to assess the performance of our approaches of unsupervised morphological segmentation and unsupervised POS tagging and show that they

achieve the state-of-the-art performance for the two morphology tasks when evaluated on a large set of languages of different typologies: analytic, fusional, agglutinative and synthetic/polysynthetic.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and above all, I praise God, the Almighty, whose blessings have made me who I am today. I thank God for providing me with this opportunity and for giving me the capability to succeed.

I would like to express my deep gratitude to my advisor Smaranda Muresan for her guidance and support along the journey. I first met her at the Center for Computational Learning Systems (CCLS) at Columbia University in 2017, while I was working as a senior research associate. She has been inspiringly successful and a role model to her students, so it was an easy decision to ask her to take me on as a PhD student, and I am so thankful she took that chance on me. Smaranda has been an exemplary advisor, and working with her is a privilege. She created a wonderful environment and gave me the chance to express my ideas and to challenge myself with interesting research directions.

I would also like to thank Owen Rambow. He was my work supervisor from 2012 until 2017. He is also a member of my proposal and thesis committees. I am grateful to know Owen, and I can only aspire to be as good a researcher, supervisor and colleague as him in the future.

I also wish to thank Michael Collins, my department advisor in the first two years of the doctorate program and a member of my proposal and thesis committees. It was an amazing experience to collaborate with such a knowledgeable and professional scientist. The success of my work would not have been complete without his guidance. I also thank Kathleen McKeown, my department advisor in the last year of the doctorate program and a member of my thesis committee. She is simply a superstar, and I am honored to work with her. I also send thanks to Daniel Bauer, my instructor in NLP and a member of my proposal and thesis committees. He has always been

supportive, and what I learned from him is priceless.

I would also like to thank all my instructors at Columbia University for their support and all the time they put into reviewing my work and providing useful feedback. I also wish to thank Doug Oard, Marine Carpuat, Judith Klavans and Maria Polinsky, from the University of Maryland, for their collaboration and fruitful insights.

I also wish to thank Tom Lippincott, who started and motivated the work on unsupervised morphological segmentation using Adaptor Grammars at the Center for Computational Learning Systems (CCLS) at Columbia University. I would also like to thank Mohammad Sadegh Rasooli, whose work on unsupervised cross-lingual dependency parsing, as part of his doctorate program at Columbia University, motivated my research on unsupervised cross-lingual part-of-speech tagging. Also, special thanks goes to Cass Lowry, from the City University of New York (CUNY), who contributed to the creation of the gold-standard datasets for Georgian, and Shinichiro Fukuda, from the University of Hawaii, who contributed to the creation of the gold-standard dataset for Japanese.

I wish to express my deepest gratitude to Nizar Habash. I would need a whole book to tell how thankful I am to know and work with him. He is my all-time role model, and without him I would not have been where I am. He is a supervisor, a friend and a brother.

Throughout my years at Columbia University, I have been fortunate to meet and work with many colleagues who have made my time at Columbia University a memorable one and have not hesitated to provide support when I needed it. I especially would like to thank Yassine Benajiba, Matt Connelly, Mona Diab, Raymond Hicks, Efsun Kayi, Rebecca Passonneau, Axinia Radeva, Ryan Roth, Ansaf Salleb-Aouissi, Rohan Shah, Nadi Tomeh, Apoorv Agarwal, Tariq Alhindi, Sakhar AlKhereyf, Mohamed Altantawy, Tuhin Chakrabarty, Ahmed El Kholy, Mohamed ElBadrashiny, Heba Elfardy, Noura Ferra, Chris Kedzie, Vinodkumar Prabhakaran, Wael Salloum and Víctor Soto. I also send thanks to the students I supervised for their hard work and creative ideas, especially Francesca Callejas, Kilol Gupta, Sujay Khandagale and Kartikeya Upasani.

I am sincerely grateful to my family, who has provided unconditional love, support and encouragement throughout the journey. A most special thanks goes to my wife Tervina, my

daughter Katelyn, my son Kevin, my parents and my parents-in-law. They are the main reason behind being able to proceed successfully.

Dedication

I dedicate my dissertation work to my wife Tervina, my daughter Katelyn, my son Kevin and any possible future children.

Chapter 1

Introduction

> "Desire is the starting point of all achievement, not a hope, not a wish, but a keen pulsating desire, which transcends everything." — *Napoleon Hill*

1.1 Overview

The majority of world's languages do not have annotated datasets, even for the basic Natural Language Processing (NLP) tasks, such as morphological segmentation and part-of-speech (POS) tagging, which in turn serve as the basis for several downstream applications, such as machine translation, speech recognition, information extraction and question answering. However, the supervised-learning approach is not always applicable since obtaining labeled data is costly and time consuming, and annotators who are native speakers might not even be available. As a result, semi-supervised and unsupervised techniques have been receiving increasing interest, especially with the recent focus on tackling linguistic diversity as technology has become accessible around the globe.

In this thesis, we present novel computational approaches for two morphology tasks: unsupervised morphological segmentation and unsupervised cross-lingual POS tagging. We evaluate our models on a large set of diverse languages across the typology spectrum, from analytic and fusional languages to agglutinative and polysynthetic ones. We also develop minimally supervised techniques that benefit from linguistic priors, when available. In addition, we propose a new architecture in which we utilize morphological segmentation to improve POS tagging, especially

for morphologically complex low-resource languages, where working in the stem/morpheme space helps derive less sparse and more efficient POS-tagging models.

1.2 Unsupervised Morphological Segmentation

Morphological segmentation is the task of breaking words into morphemes/morphs, the smallest meaningful units in a language that cannot be further divided. It is an essential subtask in many NLP applications, such as machine translation (Nguyen et al., 2010; Ataman et al., 2017) and speech recognition (Narasimhan et al., 2014). Morphological segmentation helps reduce model sparsity by operating in the morpheme/morph space. In addition, it helps recognize out-of-vocabulary words by recognizing the formation the underlying words. This is beneficial when processing low-resource languages with rich morphology. Figure 1.1 shows two morphological-segmentation examples for Arabic (upper part) and Amharic (lower part). The examples correspond to verse *MAT 15:35* in the Bible, *"He commanded the multitude to sit down on the ground"*.

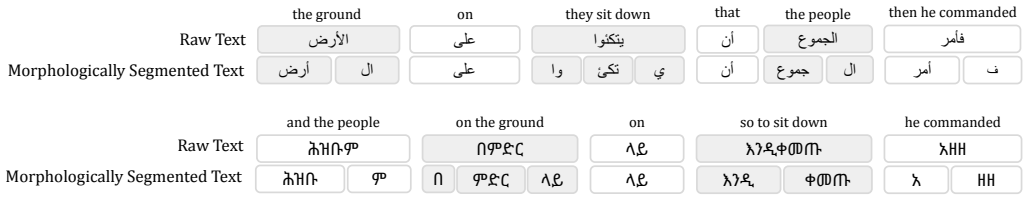


Figure 1.1: Two morphological segmentation examples for Arabic (upper part) and Amharic (lower part). Arabic reads right to left.

Since most languages lack adequate morphologically annotated resources, a number of publicly available frameworks for unsupervised and semi-supervised morphological segmentation have been developed. They rely on either generative models, such as Morfessor (Creutz and Lagus, 2007; Grönroos et al., 2014), or discriminative ones, such as MorphoChain (Narasimhan et al., 2014). A recent generative approach for morphological segmentation is based on Adaptor Grammars (AGs) (Johnson et al., 2007). 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, where probabilistic context-free grammars (PCFGs) extend this model by associating a probability to each context-free rewrite rule. AGs are Bayesian models that generalize PCFGs by weakening their independence assumption using stochastic processes called adaptors into the procedure for generating structures. In the case of morphological segmentation, a PCFG models word structure, while an adaptor adapts the subtrees and their probabilities to the corpus they are generating and acts as a caching model. For inference, AGs use a Metropolis-within-Gibbs or a Hybrid MCMC sampler (Robert and Casella, 2013) that resembles the parse tree for each input word by constructing a PCFG approximation.

In this thesis, we present *MorphAGram* [1], an AG-based framework for unsupervised and minimally supervised morphological segmentation (Eskander et al., 2020a). We derive several grammars that model word structure given language-independent specifications and specify three learning settings (Eskander et al., 2016): 1) a *Standard* setting that is fully unsupervised; 2) a *Scholar-Seeded* setting where affixes can be gathered from language resources and seeded into the grammars prior to sampling; and 3) a *Cascaded* setting that is based on self-training by deriving affixes in the *Standard* setting and then seeding them into a second round of learning. In addition, since there is no specific grammar that works well across all languages, we propose an approach that picks a nearly optimal configuration (a learning setting and a grammar) for a given unseen language (Eskander et al., 2018). We also introduce new methods for incorporating linguistic priors in the form of either designing a language-specific grammar or seeding high-quality affixes provided by an expert in the language of interest (Eskander et al., 2021). Finally, we examine multilingual training, in which we combine the lexicons of multiple related languages in low-resource setups. We test our approaches on several languages of diverse typologies in different setups, ranging from high-resource setups using Indo-European languages to low-resource setups using polysynthetic languages (Eskander et al., 2019).

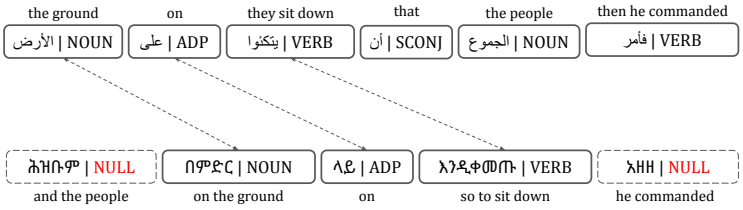
[1] https://github.com/rnd2110/MorphAGram

1.3 Unsupervised Cross-Lingual Part-of-Speech Tagging

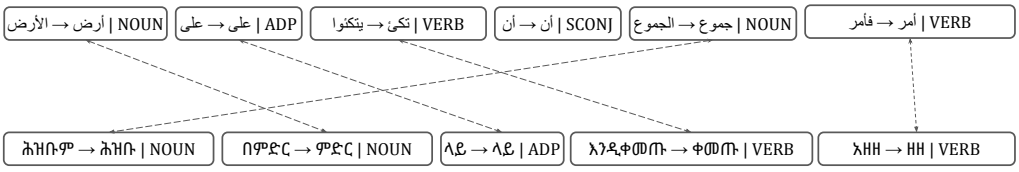
Part-of-speech (POS) tagging is the process of assigning one of the parts of speech to each word in a given text. While POS annotations are only available for a small set of languages, most of which are high-resource, efforts in documenting low-resource languages often contain translations, usually of religious text, into other high-resource languages. One such parallel corpus is the Bible (Mayer and Cysouw, 2014): 484 languages have a complete Bible translation, while 2551 have a part of the Bible translated. Translations may also be available for other materials such as movie scripts and user manuals. One popular approach to performing cross-lingual POS tagging is to harness parallel corpora by projecting POS annotations from a high-resource language for which a POS tagger is available onto a low-resource language that lacks POS-labeled data. Another approach is to perform zero-shot model transfer by tagging texts in the target language by applying a POS model of some other language, typically a related one. However, the efficiency of this approach highly relies on the relatedness between the source and target languages (Pires et al., 2019).

Unsupervised cross-lingual POS tagging via annotation projection has a long research history (Yarowsky et al., 2001; Fossum and Abney, 2005; Das and Petrov, 2011; Das and Petrov, 2011; Duong et al., 2013; Agić et al., 2015; Agić et al., 2016; Buys and Botha, 2016). These approaches either use large and/or domain-specific parallel data or rely on a large number of source languages for projection. However, since projection could suffer from bad translation, alignment mistakes or wrong assumptions, which could result in null alignments and noisy and unreliable annotations, a key consideration for all these approaches is how to obtain high-quality training instances in the target language, i.e., sentences with accurate POS tags projected from the source-language(s). Projecting from multiple languages (Fossum and Abney, 2005; Agić et al., 2015; Agić et al., 2016), graph-based label propagation (Duong et al., 2013), self-training and revision (Duong et al., 2013; Agić et al., 2016) and coupling token and type constraints (Täckström et al., 2013; Buys and Botha, 2016) are all approaches that have shown to lead to training instances of better quality. However, only one or two of these have been usually employed. Figure 1.2a shows an illustrative example of

alignment and projection from Arabic onto Amharic, where null alignments lead to null projected annotations in two Amharic words. The example corresponds to verse *MAT 15:35* in the Bible.



(a) Word-based alignment and projection



(b) Stem-based alignment and projection

Figure 1.2: An example of alignment and projection from Arabic onto Amharic. Arabic reads right to left.

In this thesis, we present a framework for unsupervised cross-lingual POS tagging via annotation projection in truly low-resource scenarios [2] (Eskander et al., 2020b), where only a limited and possibly out-of-domain set of translations into one or more high-resource languages is available. We standardize the different phases of the pipeline by integrating and expanding the best practices in alignment, projection and training, where we exploit non-contextualized and transformer-based contextualized word embeddings, affix embeddings and word-cluster embeddings within a rich neural architecture. We also propose different weighted maximum-voting, Bayesian-inference and hybrid setups for exploiting multiple sources, either in the projection phase or at decoding, as translations might be available for more than one source language. Finally, we exploit our work on unsupervised morphological segmentation to improve POS tagging by using the stem/morpheme as the core unit of abstraction for alignment and projection in order to handle low-resource languages with rich morphology in an efficient manner. Figure 1.2b shows an illustrative example of stem-

[2] https://github.com/rnd2110/unsupervised-cross-lingual-POS-tagging

based alignment and projection from Arabic onto Amharic, where the stems are based on the morphological segmentation in Figure 1.1. As illustrated, the use of the stem as the core unit of abstraction eliminates the null alignments and their corresponding null projected annotations on the Amharic side.

1.4 Evaluation and Analysis

We conduct comprehensive evaluation and analysis for our work on unsupervised morphological-segmentation and cross-lingual POS tagging and show that our approaches yield the state-of-the-art performance on both tasks. For morphological segmentation, we experiment on 13 typologically diverse languages: analytic (English), fusional (German and Arabic) agglutinative (Turkish, Finnish, Estonian, Zulu and Japanese) and synthetic/polysynthetic (Georgian, Mexicanero, Nahuatl (Mexicano), Wixarika (Huichol) and Mayo (Yorem Nokki)). We also examine low-resource scenarios of restricted data in the case of morphologically complex polysynthetic languages. In addition, we compare *MorphAGram* to state-of-the-art systems, study learning curves and analyze the segmentation outputs in all the experimental languages.

We evaluate our POS-tagging framework on 84 language pairs that belong to six source languages (English, Spanish, French, German, Russian and Arabic) and 14 target languages of diverse typologies (Afrikaans, Amharic, Basque, Bulgarian, Finnish, Georgian, Hindi, Indonesian, Kazakh, Lithuanian, Persian, Portuguese, Telugu and Turkish). We evaluate both the overall performance of the system and the performance on open-class tags. In addition, we study ablation setups of restricted data and/or computational resources, compare to state-of-the-art systems and specify the supervised setups that approximate the performance of the unsupervised ones. We also show that annotation projection outperforms zero-shot model transfer when the source and target languages are less related.

Finally, as part of our evaluation, we introduce two gold-standard morphological-segmentation datasets for Japanese and Georgian and a gold-standard POS-labeled dataset for Georgian.

1.5 Our contribution

1. We introduce a publicly available framework for unsupervised and minimally supervised morphological segmentation, *MorphAGram*, based on Adaptor Grammars (AGs). In *MorphAGram*, we define several language-independent grammars that model word structure based on different characteristics and introduce three learning settings for AGs: 1) *Standard*, with no linguistic knowledge; 2) *Scholar-Seeded*, with seeded scholar knowledge in the form of affixes generated from language resources, possibly by someone who may have never studied the underlying language; and 3) *Cascaded*, a self-training approach that approximates the benefits of linguistic knowledge for morphological segmentation in a fully unsupervised manner. In addition, since there is no single grammar that works best across all languages, we propose an approach that predicts a nearly optimal configuration (a learning setting and a grammar) for the morphological segmentation of an unseen language.

2. We propose new approaches to incorporate linguistic knowledge, when available, as priors in the segmentation models in the forms of 1) a grammar definition, through the design of a grammar that models language-specific characteristics; and 2) linguist-provided affixes, through seeding high-quality affixes compiled by an expert in the underlying language. In addition, we examine multilingual morphological segmentation, in which we combine the lexicons of multiple related languages for joint training in low-resource setups.

3. We standardize the process of annotation projection in a robust approach that exploits and expands the best practices in the literature, where we aim at producing reliable annotations of the proper quality needed to train an efficient POS tagger. This includes, but is not limited to, the use of bidirectional alignments, coupling token and type constraints on the target side and scoring the annotated sentences for the selection of reliable training instances. For training a POS model based on the projected annotations, we design a rich BiLSTM (Hochreiter and Schmidhuber, 1997) neural architecture that combines non-contextualized and transformer-based contextualized word embeddings, affix embeddings and word-cluster embeddings,

along with special handling for the null assignments resulting from missing and rejected alignments or non-overlapping token and type constraints. To our knowledge, this is the first work that exploits transformer-based contextualized word embeddings in unsupervised POS tagging.

4. We introduce new approaches for exploiting multiple source languages: 1) multi-source projection, where we combine the tags projected from multiple source languages onto the target side prior to training the POS model; and 2) multi-source decoding, where we combine the tags produced by multiple single-source models to tag a given text in the target language. Our multi-source approaches are either based on weighted maximum voting or Bayesian inference that constructs confusion matrices to learn what sources to rely on for specific sets of tags. We also conduct weighted Bayesian inference, in which we combine both mechanisms in hybrid setups. This makes a total of eight multi-source setups.

5. We combine our work on unsupervised morphological segmentation and unsupervised cross-lingual POS tagging by introducing unsupervised stem-based cross-lingual POS tagging via annotation projection, which relies on the stem as the core unit of abstraction. We use our morphological-segmentation framework *MorphAGram* to derive the stems on the target side and conduct both the alignment and projection in the stem space. In addition, we examine the use of the morpheme as the core unit of abstraction for alignment and projection, which allows for abstracting away from how the morphemes are combined in the source and target languages (e.g., whether they are free-standing or not), which is beneficial when projecting from a source language with rich morphology. Moreover, we examine the use of linguistic priors in morphological segmentation in order to improve stem-based alignment and projection towards better POS models. We also examine the use of segmentation information (stems and affixes) as learning features in our neural architecture for POS tagging. To our knowledge, this is the first work that exploits the stems and morphemes for unsupervised cross-lingual POS tagging.

6. We perform extensive evaluation and analysis for our morphological-segmentation and POS-tagging frameworks. In the case of morphological segmentation, we evaluate *MorphAGram* on 13 languages of diverse typologies within high-resource and low-resource setups, where we introduce two new gold-standard morphological-segmentation datasets for Japanese and Georgian. In the case of POS tagging, we evaluate our models on six source languages and 14 target languages of diverse typologies, for a total of 84 language pairs, where we introduce a new gold-standard POS-labeled dataset for Georgian.

1.6 Thesis Outline

This thesis is organized as follows: Chapter 2 briefly overviews related work on unsupervised morphological segmentation and unsupervised cross-lingual POS tagging. In Chapter 3, We discuss our work on unsupervised morphological segmentation, where we present the *MorphoAGram* framework along with the incorporation of linguistic priors and multilingual training. The next three chapters, from 4 to 6, are dedicated to our work on unsupervised cross-lingual POS tagging. Chapter 4 presents our POS-tagging architecture using a single source language, while Chapter 5 presents our multi-source approaches to exploit multiple source languages using weighted maximum-voting, Bayesian-inference and hybrid setups. Chapter 6 then presents an end-to-end approach in which we combine our work on unsupervised morphological segmentation and unsupervised cross-lingual POS tagging for unsupervised stem-based cross-lingual POS tagging for low-resource morphologically complex languages, where we use the stem as the core unit of abstraction. We also present morpheme-based alignment and projection, the use of linguistic priors towards better POS models and the use of segmentation information as learning features. Finally, we conclude and discuss possible future directions in Chapter 7.

Chapter 2

Related Work

"No man becomes rich without himself enriching others." — *Andrew Carnegie*

2.1 Unsupervised Morphological Segmentation

2.1.1 A Glimpse of History

Morphological segmentation was first performed by applying manual rule engineering, which is costly and time consuming. It also requires extensive linguistic knowledge, which might not be accessible for several low-resource and endangered languages. The use of finite state automata (FSA) was then widely investigated for morphological segmentation when labeled data is not accessible (Koskenniemi, 1984; Johnson and Martin, 2003; Goldsmith and Hu, 2004). In this classical approach, FSA are used to describe the possible word forms of a language given a lexicon of words. One major concern is that FSA are not capable of generalizing to unseen words and irregular structures, which makes them not suitable to process morphologically complex languages of rich affixation and templatic morphology. Moreover, FSA are less efficient when applied in an unsupervised manner without access to labeled data.

Another variation of the automaton approach is modeling the words as suffix trees, where the nodes that have identical continuations can be compressed. Segmentation boundaries can then be suggested where a node has a high branching factor, which represents a location where the next letter has a low predictability (Harris, 1970; Déjean, 1998).

The task of unsupervised morphological segmentation received more focus with the early

advances in machine learning, powered by the increasing interest in low-resource languages. Kazakov (1997) and Goldsmith (2001) proposed the utilization of the minimum description length (MDL) principle for unsupervised morphological segmentation, where the objective is to find the most compact form of segmentation given an input corpus. However, the approach is hard to generalize across out-of-vocabulary words and across different languages.

The success of the MDL principle in the task of unsupervised morphological segmentation was the trigger for several unsupervised generative models that are bundled together as the *Morfessor* family (Creutz and Lagus, 2007; Grönroos et al., 2014) (Section 2.1.2). Another generative approach is based on Adaptor Grammars (AGs) (Johnson et al., 2007) (Section 2.1.3), where probabilistic grammars are utilized to model word structure. In parallel, discriminative log-linear models have been developed for the task of unsupervised morphological segmentation (Poon et al., 2009; Narasimhan et al., 2015) (Section 2.1.4). However, while the discriminative models are more efficient at handling larger amounts of available data than their generative counterparts (Ruokolainen et al., 2016), the generative models learn better from small datasets and better allow for the incorporation of linguistic priors as additional signals in minimally supervised learning setups.

In addition to the unsupervised approaches, supervised morphological segmentation has been widely investigated. For instance, Ruokolainen et al. (2013) proposed a CRF approach that is efficient at learning from small labeled data, while Kann et al. (2018) proposed different neural approaches for the morphological segmentation of polysynthetic languages, where they experimented with data-augmentation and joint-training setups. Another neural approach was proposed by Ansari et al. (2019), in which they exploited rich annotated lexicons.

2.1.2 The Morfessor Family

Morfessor is a commonly used framework for unsupervised and semi-supervised morphological segmentation. At first, Creutz and Lagus (2002) proposed two baseline segmentation approaches that utilize an input corpus of plain text. The first approach is based on recursive segmentation,

where it examines all the possible ways to segment each word in a recursive manner. It then assigns the optimal segmentation that minimizes the segmentation cost based on the MDL principle, where the cost is defined to be proportional to the number of morphs in the induced segmented corpus and lexicon of morphs. The second approach is based on sequential segmentation, where it assigns each word a random segmentation and then iteratively computes the probabilities of the morphs. It then utilizes the Viterbi algorithm to find the lowest maximum-likelihood cost.

Morfessor Baseline-Length was then introduced by Creutz (2003), which is a generative model of two main stochastic processes. The first process generates a lexicon of morphs, while the second one generates a corpus that is a sequence of morphs. The segmentation is performed by generating a corpus that has exactly the same sequence as the input corpus of plain text, where the recursive segmentation approach by Creutz and Lagus (2002) is utilized to find the optimal segmentation. However, two priors are needed in order to optimize the segmentation: 1) the most common morph length; and 2) the proportion of morph types that appear only once.

Creutz and Lagus (2004) then introduced *Morfessor Categories-ML*. The model is a maximum-likelihood (ML) model that analyzes the segmentation induced by the *Baseline-Length model*. The model learns the categories of the morphs and the dependencies between these categories, where the left and right character-level contexts are taken into consideration.

Morfessor Categories-MAP was subsequently introduced by Creutz and Lagus (2005a). The system utilizes a probabilistic maximum-a-posteriori (MAP) model that builds hierarchical representations of the morphs. The induced lexicon of morphs stores parameters that are related to the meaning and formation of the morphs.

The different models of *Morfessor* were then combined together, and the first release was made publicly available (Creutz and Lagus, 2005b; Creutz and Lagus, 2007). A Python version was then released at a later stage by Virpioja et al. (2013).

Morfessor FlatCat was introduced by Grönroos et al. (2014). The system is based on *Morfessor Categories-ML* and *Morfessor Categories-MAP*, but it uses a flat lexicon that allows for learning from labeled data in a semi-supervised manner, in addition to unsupervised learning. *Morfessor*

FlatCat was made publicly available [1] by Smit et al. (2014).

Grönroos et al. (2014) reported BPR (Boundary Precision and Recall) F1-scores of 65.0%, 61.0% and 62.0% for English, Finnish and Turkish, respectively, using *Morfessor Categories-MAP*, and BPR F1-scores of 69.0%, 52.0% and 51.0% for English, Finnish and Turkish, respectively, using *Morfessor FlatCat*, where BPR measures the ability of the system to discover segmentation boundaries (Virpioja et al., 2011). We use *Morfessor* as a baseline in the evaluation of our morphological-segmentation framework, *MorphaGram*, and report its performance on our experimental languages.

2.1.3 Adaptor Grammars

Adaptor Grammars (AGs) (Johnson et al., 2007) are nonparametric Bayesian models that learn distributions over trees by generalizing probabilistic context-free grammars (PCFGs). An AG is composed of two main components; a PCFG and an adaptor. The PCFG models the entity of interest; that is, word structure in the case of morphological segmentation, while the adaptor is a component that is based on the Pitman-Yor process (Pitman, 1995) and adapts the subtrees and their probabilities to the corpus they are generating. It also acts as a caching model. For inference, AGs use a Metropolis-within-Gibbs or a Hybrid MCMC sampler (Robert and Casella, 2013) that infers the posterior distribution over the trees and all the hyperparameters of the model.

Johnson (2008a) explored AGs for the tasks of unsupervised word segmentation and unsupervised morphological segmentation for Sesotho, where the grammars model sentence structure and word structure, respectively. Johnson (2008a) reported morpheme-based F1-scores up to 39.3%. However, the main takeaway is that there is a vast variation in how the different grammars perform. In addition, more structured grammars achieve better segmentation than simpler ones.

Botha and Blunsom () extended AGs to model non-concatenative morphology, such as infixation, circumfixation and root-templatic derivation. They replaced the PCFGs by PSRCGs (probabilistic simple-range concatenating grammars). The components of a PSRCG are the same as those of a

[1] https://morfessor.readthedocs.io/en/latest/index.html

PCFG except that nonterminals accept arguments (variables), where a variable that appears in a production rule must be used exactly once on the left and the right sides. A nonterminal becomes instantiated when its variables are bound to ranges through substitution. The approach achieves BPR F1-scores up to 74.5% and 78.1% for Arabic and Hebrew, respectively.

Another work that utilizes AGs for morphological segmentation was proposed by Sirts and Goldwater (2013), where they defined and compared different grammars to model word structure. They also experimented with three setups: 1) Unsupervised: no labeled data; 2) *AG ssv*: a semi-supervised approach in which small labeled data is used to extract counts of grammar rules and subtrees in order to guide the sampler that operates on both the labeled and unlabeled data; and 3) *AG Select*: a morphological template (metagrammer) that is a binary tree of four levels and is used to discover the best grammar (subtree) using small labeled data. The different setups were evaluated on English, Finnish, Estonian and Turkish, where the unsupervised setup achieves BPR F1-scores of 66.1%, 67.5%, 61.6% and 61.1%, respectively, while *AG ssv* achieves BPR F1-scores of 70.5%, 69.7%, 70.0% and 70.3%, respectively, and *AG Select* achieves BPR F1-scores of 69.8%, 68.8%, 67.5% and 70.1%, respectively.

In the evaluation by Botha and Blunsom () and Sirts and Goldwater (2013), AG proved successful, where they significantly outperform *Morfessor* in the cases of Arabic, Hebrew and Estonian by average relative error reductions of 56.2%, 71.7% and 33.5%, respectively.

Our work on unsupervised morphological segmentation is based on AGs, where we define a large number of language-independent grammars, define different learning settings that either are fully unsupervised or exploit linguistic knowledge, derive new approaches for the automatic tailoring of grammars for unseen languages and incorporate linguistic priors in a minimally supervised manner. In addition, we examine multilingual joint-training using related languages in low-resource setups. We compile our grammars and modules under the *MorphaGram* framework and evaluate it on 13 languages of diverse typologies.

2.1.4 Log-Linear Discriminative Models

In parallel to the generative models described above, a number of log-linear discriminative models were introduced for the task of unsupervised morphological segmentation and proved successful.

Log-linear models were first introduced for unsupervised morphological segmentation by Poon et al. (2009). They used character-level bigrams and trigrams in addition to morpheme-level global features to model word structure. The model incorporates exponential priors inspired by the MDL principle along with contrastive estimation and sampling for learning and inference. The system achieves BPR F1-scores of 78.1% and 66.9% for Arabic and Hebrew, respectively. One drawback is that the system has a high degree of computational complexity, where it is suggested to impose linguistic assumptions in order to reduce complexity.

Another log-linear model is *MorphoChain* [2] (Narasimhan et al., 2015). In *MorphoChain*, words are modeled as morphological chains, where a morphological chain is a short sequence of words that starts with a base word (a parent) and ends with a morphological variant, e.g., $port \rightarrow report \rightarrow reporting$. The model is a log-linear feature-based hidden Markov model (HMM) that predicts the parent of a given word, where the segmentation is assigned by tracing the changes in the chain until the parent is reached. The model utilizes several features that indicate the transformations across the morphological chains, e.g., insertions, deletions and repetitions, along with affix identity, suffix correlation and embedding-based similarity. The model achieves BPR F1-scores of 76.2%, 61.2% and 79.9% for English, Turkish and Arabic, respectively. However, the performance drops significantly when using small datasets, which is not suitable for truly low-resource setups. We use *MorphoChain* as a baseline in the evaluation of our morphological-segmentation framework, *MorphaGram*, and report its performance on our experimental languages.

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

2.2 Unsupervised Cross-Lingual Part-of-Speech Tagging

2.2.1 A Glimpse of History

Part-of-speech (POS) tagging was first performed in a rule-based fashion, such as the work performed by Greene and Rubin (1971), where they constructed rules to tag the Brown corpus [3]. Handcrafting rules requires linguistic knowledge that might not be available in truly low-resource scenarios. In addition, it involves a significant overhead in terms of time and cost.

Since labeling data for POS tags is a time-consuming and expensive process, while annotators might not even be available for some languages, several semi-supervised approaches have been employed for POS tagging without reliance on annotated texts. A common formulation of a minimally supervised POS tagger takes the form of an HMM in which the emission, $P(w_{\text{i}}|t_{\text{i}})$, and transition, $P(t_{\text{i}}|t_{\text{i-1}}...t_{\text{i-n}})$, probabilities are estimated from a lexicon that contains POS information. An example is the work by Banko and Moore (2004), where they derived the HMM parameters by constructing a lexicon of POS information based on the English Penn Treebank [4] (Marcus et al., 1993) and applied different techniques for noise reduction. Another example is the work by Li et al. (2012), in which they used the Wiktionary [5], large-scale, continuously growing and high in coverage, to build HMM taggers. Another approach is joint learning, which assumes access to small labeled data that can be combined with additional labeled data of one or more related languages in a multilingual learning setup that exploits the common space shared across the training languages. An example is the work by Cotterell and Heigold (2017), where they combined the character embeddings of the target language with those of a related high-resource language and experimented with different setups in which the source and target languages have either separate or common output layers.

During the last two decades, there has been increasing interest in developing fully unsupervised POS taggers that assume no access to language resources containing POS information nor labeled

[3] https://archive.org/details/BrownCorpus
[4] https://catalog.ldc.upenn.edu/LDC2015T13
[5] http://wiktionary.org

data in the language of interest. This is necessary to process several low-resource and endangered languages. It also speeds up the development of NLP resources for such languages as there is no need to develop tailored taggers of language-specific information/data for each language.

There are two main approaches that have been proposed for fully unsupervised POS tagging: 1) cross-lingual POS tagging via annotation projection (Section 2.2.2); and 2) cross-lingual POS tagging via zero-shot model transfer (Section 2.2.3). Both approaches rely on the existence of another language, a source language, for which a POS model is accessible. In the annotation-projection approach, the tags in the source are projected onto the target through a parallel text, while in the model-transfer approach, the source model is applied directly on the target text.

2.2.2 Cross-Lingual Part-of-Speech Tagging via Annotation Projection

Unsupervised cross-lingual POS tagging via annotation projection assumes access to some parallel text between the target language and a source one for which a POS tagger is accessible, which is used to tag the text on the source side. First, a word-based alignment model is trained based on the parallel text and is used to induce the word-level alignments between the source and target sides. The source tags are then projected onto the target across the word-level alignments and become the basis to train a POS tagger for the target language.

Unsupervised cross-lingual POS tagging via annotation projection was first introduced by Yarowsky et al. (2001), where they applied noise-reduction and smoothing techniques to process the potentially wrong and null alignments. They then used the induced transition and emission probabilities on the target side to train an HMM POS tagger. They further extended their work for other NLP tasks, namely noun-phrase bracketing, named-entity recognition and lemmatization. They reported a 76.0% POS-tagging accuracy when projecting from English to French.

In addition to the noise-reduction and smoothing techniques by Yarowsky et al. (2001), several approaches have been proposed to improve the projected annotations as they suffer from several issues such as bad translation, alignment mistakes and inconsistencies between languages. These techniques are 1) multi-source projection; 2) graph-based label propagation; 3) self-training and

revision; and 4) coupling token and type constraints.

Multi-source projection was first introduced by Fossum and Abney (2005), where they combined the outputs of single-source taggers based on different source languages through either maximum voting or linear combination that is based on the tag distribution of the underlying word types per source language. The approach achieves accuracies up to 89.8% and 67.4% when evaluated on French and Czech, respectively, through multi-source projection from English, German and Spanish.

In efforts to increase the coverage of the projected annotations, Das and Petrov (2011) proposed graph-based label propagation to expand the projected tags on the target side. The induced distributions are then used to construct a log-linear feature-based HMM with L-BFGS, a quasi-Newton method, optimization (Liu and Nocedal, 1989). They achieved an average accuracy of 83.4% on eight Indo-European languages, namely Danish, Dutch, German, Greek, Italian, Portuguese, Spanish and Swedish. Alternatively, Duong et al. (2013) applied self-training and revision, where the implemented tagger is used to fill in the POS gaps in the annotated text of the target language, which are mainly due wrong and null alignments. The probabilities are then recalculated and become the basis for a new iteration. They achieved the same average accuracy of 83.4% as the approach by Das and Petrov (2011), on the same evaluation sets, but using a model of considerably less complexity.

Agić et al. (2015) combined multi-source projection with self-training and revision, where they utilized a large number of source languages in a bootstrapping setup. The approach is to first project the annotations from k source languages onto $n - k$ target languages to build initial taggers that are then used to fill in the gaps in the annotated texts of the target languages. Then, for each target language, a new tagger is developed based on the projected tags from the remaining $n - 1$ languages. They showed that bootstrapping helps in 16 out of 25 target languages. The best results are achieved in the cases of Portuguese and Spanish, with accuracies of 83.8% and 81.4%, respectively.

Agić et al. (2016) then improved the multi-source projection approach by Agić et al. (2015) by weighting the projected tags based on the probabilities of the corresponding alignments along with

the use of Efmaral [6] (Östling et al., 2016) for word-level alignments (instead of Fast_Align [7] (Dyer et al., 2013)) and the Watchtower Corpus (WTC) [8] as the source of parallel data (instead of the Bible). With the elimination of bootstrapping, they achieved relative error reductions of 3.7% and 26.9% for Portuguese and Spanish, respectively.

Täckström et al. (2013) improved the projection of the tags by coupling token and type constraints, where the token constraints represent the projected tags, while the type constraints represent the tag distribution of each word type on the target side and are used to control the accepted token constraints. They achieved an average accuracy of 84.9% on the same evaluation sets used by Das and Petrov (2011) and Duong et al. (2013). Similarly, Buys and Botha (2016) coupled token and type constraints, where the type constraints are assigned to those tokens of missing token constraints. The coupled constraints are then used in a Wsabie neural model (Weston et al., 2011) that learns to rank the set of tags allowed by the coupled constraints. They achieved an average accuracy of 80.1% on 11 Indo-European languages.

Some work on annotation projection exploits an existing lexicon of POS information in a semi-supervised manner. For instance, Das and Petrov (2011) showed that extracting tagging dictionaries from the treebanks and using them as constraint features in the feature-based HMM results in an average error reduction of 62.0%. Also, Täckström et al. (2013) showed that the use of the Wiktionary to define the type constraints improves the performance by a relative error reduction of 3.3%. Another semi-supervised approach that relies on annotation projection was proposed by Plank and Agić (2018), where they utilized Polyglot embeddings (Al-Rfou' et al., 2013) and lexical information from the Wiktionary within the approach proposed by Agić et al. (2016). They showed significant error reductions of 50.6% and 41.2% for Portuguese and Spanish, respectively.

On another hand, Fang and Cohn (2016) proposed a distant-supervision approach in which they trained a BiLSTM (Hochreiter and Schmidhuber, 1997) POS model on 1,000 manually annotated words. They then trained another POS model on both the manually annotated words and words with

[6] https://github.com/robertostling/efmaral
[7] https://github.com/clab/fast_align
[8] https://www.jw.org/en/online-help/watchtower-library

automatically projected tags, where the tag probabilities from the supervised model are adjusted into a distribution that matches the projected tags (bias transformation) through the use of a confusion matrix that models the correspondence between the manual tags and the projected ones. They achieved an average accuracy of 91.7% on the same evaluation sets used by Das and Petrov (2011), Duong et al. (2013) and Täckström et al. (2013). They also showed that coupling the manual and projected tags prior to training the POS model underperforms the use of only the projected tags.

Another work that is based on annotation projection in a semi-supervised manner is the work by Cucerzan and Yarowsky (2002), in which a bilingual dictionary is used to extract the tag distributions of the words in the target language. However, the handling of inflected words and closed-class words relies on manual paradigms and hand-crafted rules, while irregularities and out-of-vocabulary words are handled through edit-distance measurements, which makes the approach costly and time consuming, along with the need to access extensive linguistic knowledge.

While most prior work on POS tagging via annotation projection does tagging for several target languages, some research focuses on specific language pairs, such as the projection from Russian to Ukrainian in a fully unsupervised manner (Huck et al., 2019) and the projection from German to Hittite using distant supervision (Sukhareva et al., 2017).

Our work on unsupervised cross-lingual POS tagging is based on annotation projection, where we exploit and expand the best practices in the literature in order to produce reliable annotations towards efficient neural POS models that combine word embeddings, affix embeddings and word-cluster embeddings. Moreover, we derive new approaches for multi-source projection and decoding, in addition to stem-based and morpheme-based alignment and projection using our morphological-segmentation framework, *MorphAGram*. We evaluate our models using 6 source languages and 14 target languages, for a total of 84 language pairs of diverse typologies.

Finally, it is worth noting that there has been a lot of research on unsupervised and weakly supervised cross-lingual learning via annotation projection for other tasks in NLP, such as named-entity recognition (Ni et al., 2017; Ehrmann et al., 2011), dependency parsing (Tiedemann, 2014; Rasooli and Collins, 2015) and semantic role labeling (Padó and Lapata, 2009).

2.2.3 Cross-Lingual Part-of-Speech Tagging via Zero-Shot Model Transfer

In addition to conducting cross-lingual POS tagging through annotation projection, zero-shot model transfer has been widely explored. In this approach, instead of learning a POS model of the target language, a trained POS model of some other language, preferably a related one, is applied directly on texts in the target language. For instance, Huck et al. (2019) showed the effectiveness of zero-shot model transfer from Russian to Ukrainian, two highly related Slavic languages. They however obtained noticeably better results through annotation projection. Another example was proposed by Pasha et al. (2014), where they applied MADA (Habash and Rambow, 2005), a POS tagger of Modern Standard Arabic, on Egyptian Arabic, a dialect of Arabic.

Pires et al. (2019) then widely investigated zero-shot model transfer by fine-tuning the multilingual transformer-based *BERT* language model (*mBert*) (Devlin et al., 2019) for the POS tagging of a source language and applying the fine-tuned model on the target one. While the approach does not require annotations on the target side nor translations between the source and target languages, it is highly sensitive to the relatedness between the source and target languages, where it cannot generalize well across languages of different morphological typologies. For instance, while transferring from English to Bulgarian, two Indo-European languages, yields a POS-tagging accuracy of 87.1%, transferring from English to Japanese, two morphologically unrelated languages, results in a significantly lower accuracy of 49.4%.

2.3 Part-of-Speech Tagsets

While English has nine main parts of speech that are commonly taught in school, namely adjective, adverb, article, conjunction, interjection, preposition, pronoun, noun and verb, there are many fine-grained categories that are necessary to describe the different morphosyntactic features of a word, such as to distinguish between singular and plural nouns, past and present verbs, personal and demonstrative pronouns, etc. A commonly used POS tagset in English NLP in the past is the one used in the Brown corpus [9], which consists of 87 tags, including 10 nominal tags and seven verbal

[9] http://korpus.uib.no/icame/manuals/brown/index.htm

ones. The Brown tagset was then the basis for the Penn-Treebank tagset [10], in which the number of tags is reduced to 36, including four nominal tags and seven verbal ones. The Penn tagset has been widely adopted in English NLP.

However, languages differ in their morphological characteristics. For example, English POS tags are not sufficient to cover the morphological categories seen in other languages. For instance, some languages have words marked for their case (accusative, dative, genitive and nominative), and others have verbs marked for their aspect (active and passive). This was the motivation for Petrov et al. (2012) to develop a universal POS tagset [11] that abstracts away from language-specific categories while generalizing well across all languages. The universal tagset contains 12 POS tags that are based on mapping the tags in the treebanks of 22 languages. Petrov et al. (2012) demonstrated that the universal POS tags generalize well across language boundaries on an unsupervised grammar-induction task and is giving competitive parsing accuracies to those of the corresponding supervised task. The universal tagset by Petrov et al. (2012) had been widely used in cross-lingual POS tagging (Das and Petrov, 2011; Duong et al., 2013; Täckström et al., 2013; Fang and Cohn, 2016) until the development of the Universal-Dependencies (UD) POS tagset [12] as part of the UD project [13]. The UD tagset contains 17 universal tags and has become widely adopted in the NLP community (Agić et al., 2015; Agić et al., 2016; Buys and Botha, 2016; Plank and Agić, 2018; Huck et al., 2019). In this thesis, we use the UD tagset in our work on unsupervised cross-lingual POS tagging, which allows us to build efficient cross-lingual models and to compare them to the state-of-the-art approaches.

[10] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
[11] https://github.com/slavpetrov/universal-pos-tags
[12] https://universaldependencies.org/u/pos
[13] https://universaldependencies.org

Chapter 3

Unsupervised Morphological Segmentation

3.1 Overview

Morphological segmentation breaks words into morphemes/morphs, the smallest meaningful units in a language. When learning morphological segmentation given a list of words in an unsupervised manner, a reasonable objective is to find the segmentation that minimizes the size of the induced lexicon of morphemes/morphs, in order to prevent under-segmentation, and that minimizes the average number of morphemes/morphs per word, in order to prevent over-segmentation (Creutz and Lagus, 2002).

In this thesis, we focus on unsupervised morphological segmentation via generative models that are based on Adaptor Grammars (AGs) (Johnson et al., 2007). AGs are nonparametric Bayesian models that generalize probabilistic context-free grammars (PCFGs) and have proved successful for different NLP tasks including morphological segmentation, in which a PCFG models word structure, in both unsupervised and minimally supervised learning setups (Sirts and Goldwater, 2013; Botha and Blunsom,).

Our contribution is as follows:

- We introduce a publicly available AG-based framework for unsupervised and minimally supervised morphological segmentation, *MorphAGram* [1], in which we define several language-independent grammars that model word structure based on different characteristics (Section 3.3.1) and introduce three learning settings for AGs: 1) *Standard*, with no linguistic knowledge; 2) *Scholar-Seeded*, with seeded linguistic knowledge in the form of affixes compiled from language resources; and 3) *Cascaded*, a novel approach that relies on self-training

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

to approximate the benefits of linguistic knowledge for morphological segmentation in a fully unsupervised manner (Section 3.3.2). In addition, we propose an approach for the automatic tailoring of grammars, where we predict a nearly optimal configuration (a learning setting and a grammar) for the morphological segmentation of unseen languages (Section 3.3.3).

- We propose new approaches to incorporate linguistic knowledge as priors in the morphological-segmentation models in terms of 1) grammar definition, through the design of a grammar that models language-specific characteristics; and 2) linguist-provided affixes, through seeding high-quality affixes compiled by an expert in the underlying language (Section 3.3.4).

- We examine multilingual morphological segmentation, in which we combine the lexicons of multiple related languages for joint training in low-resource setups (Section 3.3.5).

- We perform extensive evaluation and analysis on 13 languages of diverse typologies within high-resource and low-resource setups. The languages are English, German, Finnish, Estonian, Turkish, Zulu, Japanese, Georgian, Arabic, Mexicanero, Nahuatl (Mexicano), Wixarika (Huichol) and Mayo (Yorem Nokki). As part of the evaluation, we introduce two new gold-standard morphological-segmentation datasets for Japanese and Georgian (Sections 3.4 and 3.5).

We show that *MorphAGram* is highly efficient for unsupervised and minimally supervised morphological segmentation for all the experimental languages, including the polysynthetic ones that we examine in truly low-resource setups. Using the BPR metric (Virpioja et al., 2011) that measures the ability of the system to recognize segmentation boundaries, our fully unsupervised setup achieves F1-scores ranging from 62.7%, for Zulu, to 82.5%, for Arabic, where the F1-scores are at least 75.0% for eight languages out of our 13 experimental ones. Moreover, seeding the affixes in the *scholar-seeded* setting achieves an average relative error reduction of 5.1%. We then compare *MorphAGram* to two strong baselines, *Morfessor* (Creutz and Lagus, 2007; Grönroos et al., 2014) and *MorphoChain* (Narasimhan et al., 2014), and show average relative error reductions of 22.8% and 40.7%, respectively, using our fully unsupervised setup. We also show significant improvements

due to the incorporation of linguistic priors through the design of a Japanese-specific grammar and through the seeding of Georgian and Arabic linguist-provided affixes. We achieve relative error reductions of 4.2%, 33.2% and 32.9% for Japanese, Georgian and Arabic, respectively, (Section 3.5). Finally, in the case of multilingual morphological segmentation, we obtain performance gains for Estonian when we combine small-sized lexicons of Finnish and Estonian, two Uralic languages, for joint training (Section 3.5.6).

This chapter contains and expands our work on unsupervised morphological segmentation using AGs, where we introduce several grammar definitions and define three main learning settings (Eskander et al., 2016), in addition to the automatic tailoring of AGs for unseen languages (Eskander et al., 2018), examining low-resource and multilingual setups using polysynthetic languages (Eskander et al., 2019) and incorporating linguistic priors in a minimally supervised learning manner (Eskander et al., 2021). The work is compiled and packaged into the *MorphAGram* framework for unsupervised and minimally supervised morphological segmentation (Eskander et al., 2020a).

3.2 Background

Adaptor grammars (AGs) (Johnson et al., 2007) are nonparametric [2] Bayesian models that generalize PCFGs by weakening their independence assumptions using additional stochastic processes called adaptors into the procedure for generating structures. In this procedure, 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 linguistic data.

AGs define a framework to implement nonparametric Bayesian learning of grammars and are usually trained in an unsupervised manner using sampling techniques. AGs have been used successfully for unsupervised and minimally supervised morphological segmentation (Sirts and Goldwater, 2013; Botha and Blunsom,). AGs have also been applied in other NLP applications, such as word segmentation (Johnson, 2008a; Johnson, 2008b; Johnson and Demuth, 2010), named-entity

[2]The term "nonparametric" means that the learning process considers models with different sets of parameters.

clustering (Elsner et al., 2009), transliteration of names (Huang et al., 2011) and native-language identification (Wong et al., 2012).

An AG is composed of two main components: a PCFG and an adaptor. In the case of morphological segmentation, the PCFG is a morphological grammar that specifies word structure, while the adaptor adapts the subtrees and their probabilities to the corpus they are generating and acts as a caching model. The adaptor is based on the Pitman-Yor process (Pitman, 1995).

An AG has a vector of concentration parameters α. A nonterminal A that has $\alpha_{\mathrm{A}} = 0$ is unadapted, meaning that A expands as in an ordinary PCFG, where a production rule $A \rightarrow \beta$ is picked with probability $p(A \rightarrow \beta)$, and β is expanded recursively. If $\alpha_{\mathrm{A}} > 0$, then A is adapted, and α_{A} becomes the Dirichlet concentration parameter associated with nonterminal A. For an adapted nonterminal A that is expanded n_{A} times before, there are two possible events.

1. A expands to subtree σ with probability $\frac{n_{\alpha}}{n_{\mathrm{A}}+\alpha_{\mathrm{A}}}$, where α_{A} is the number of times A expanded to α before.

2. A expands as an unadapted nonterminal with probability $\alpha_{\mathrm{A}}\frac{p(A\rightarrow\beta)}{n_{\mathrm{A}}+\alpha_{\mathrm{A}}}$.

Accordingly, an adapted nonterminal A either expands to a previously expanded subtree with a probability proportional to the number of times it is utilized or expands as in an ordinary PCFG with a probability proportional to the concentration parameter.

For inference, AGs use a Metropolis-within-Gibbs or a Hybrid MCMC sampler (Robert and Casella, 2013) that resembles the parse tree for each word in the input lexicon, conditioned on the parses of the other words. The algorithm constructs a PCFG approximation to the AG which contains one rule for each adapted subtree α and uses a Metropolis accept/reject step to correct for the difference between the true AG distribution and the generated PCFG approximation. Thus, the sampler is used to infer the posterior distribution over the trees and all the hyperparameters of the model, including the PCFG probabilities in the base distribution and the hyperparameters of the Pitman-Yor process. For more comprehensive details about how AGs work, see Johnson et al. (2007).

3.3 The MorphAGram Framework

In this thesis, we introduce *MorphAGram*, a publicly available framework for unsupervised and minimally supervised morphological segmentation that is based on Adaptor Grammars (AGs). In *MorphAGram*, we define several language-independent grammars and introduce different learning settings that are either unsupervised or minimally supervised. *MorphAGram* also allows for the automatic tailoring of grammars for unseen languages and for the incorporation of linguistic priors, in the form of either grammar definition or linguist-provided affixes. *MorphAGram* is also suitable for multilingual learning, in which the lexicons of related languages are combined together. We describe below the different components and modules in the *MorphAGram* framework.

3.3.1 Grammar Definitions

The first step in learning morphological segmentation using AGs is to define the grammars that model word structure. The definition of a grammar relies on three main dimensions:

- **Word Modeling**: A word can be modeled as a sequence of generic morphemes/morphs or as a sequence of a prefix, a stem and a suffix, where a nonterminal may be recursively defined to allow for compounding.
- **Level of Abstraction**: Basic elements can be combined into more complex nonterminals, e.g., *Compound*, or split into smaller ones, e.g., *SubMorph*.
- **Segmentation Boundaries**: This dimension defines the nonterminals that incur splits in the final segmentation output. For example, a word can be segmented into a complex prefix, a stem and a complex suffix (three-way segmentation), e.g., *redis+cover+ing* and *irre+place+ables*, or it can be split into a stem and simple affixes (multiway segmentation), e.g., re+dis+cover+ing and *ir+re+place+able+s*.

Table 3.1 lists nine grammars and their characteristics. *Morph+SM* and *PrStSu2a+SM* are two baseline grammars that were first introduced by Sirts and Goldwater (2013), while the rest of the

grammars are new derivations that we introduce and include in the *MorphAGram* framework. The selection of the proper grammar relies on the target language and the downstream application as the performance of the grammars differs across different languages, while a downstream application might benefit from specific nonterminals or require a specific level of granularity in the induced segmentation.

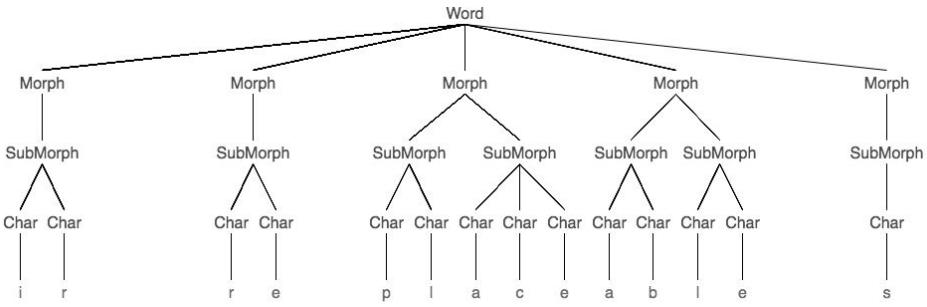
Grammar	Word Modeling	Compound	Morph	SubMorph	Segmentation Boundaries
Morph+SM	Morph		Y	Y	Morph
Simple	Prefix+Stem+Suffix				Prefix-Stem-Suffix
Simple+SM	Prefix+Stem+Suffix			Y	Prefix-Stem-Suffix
PrStSu	Prefix+Stem+Suffix		Y		PrefixMorph-Stem-SuffixMorph
PrStSu+SM	Prefix+Stem+Suffix		Y	Y	PrefixMorph-Stem-SuffixMorph
PrStSu+Co+SM	Prefix+Stem+Suffix	Y	Y	Y	Prefix-Stem-Suffix
PrStSu2a+SM	Prefix+(Stem-Suffix)		Y	Y	PrefixMorph-Stem-SuffixMorph
PrStSu2b+SM	(Prefix-Stem)+Suffix		Y	Y	PrefixMorph-Stem-SuffixMorph
PrStSu2b+Co+SM	(Prefix-Stem)+Suffix	Y	Y	Y	Prefix-Stem-Suffix

Table 3.1: Grammar definitions for modeling word structure. Y=applicable.

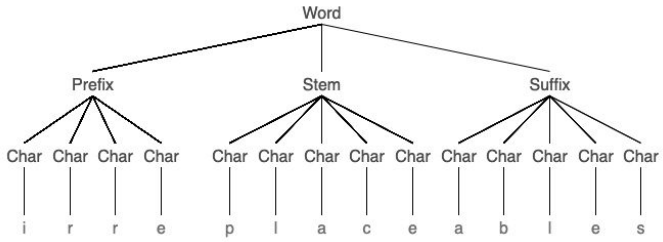
It is noteworthy to mention that we derived and experimented with other grammar variations, considering the combinations of the different characteristics, but we eliminated the grammars that do not perform well across our development languages, namely English, German, Finnish, Estonian, Turkish and Zulu (Section 3.4), and those that perform similarly to other grammars.

For word modeling, all the grammars model the word as a sequence of a prefix, a stem and a suffix except the *Morph+SM* grammar, in which the word is modeled as a sequence of morphs. In addition, in all the grammars denoted by *PrStSu*, prefixes and suffixes are recursively defined as a sequence of affix morphs in order to allow for affix compounding.

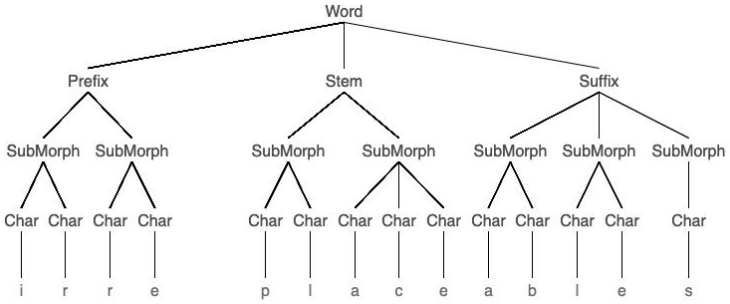
Regarding the level of abstraction, all the grammars denoted by *Co* involve a high-level nonterminal, *Compound*, that expands to a prefix, a stem or a suffix, while those denoted by *SM* involve a low-level nonterminal, *SubMorph*, that expands to a sequence of characters. These nonterminals allow prefixes, stems and suffixes to share common information, which is efficient for languages of rich affixation. We use the terms *2a* and *2b* to denote binary high-level nonterminals that combine stems with suffixes (*Stem-Suffix*) and prefixes with stems (*Prefix-Stem*), respectively.



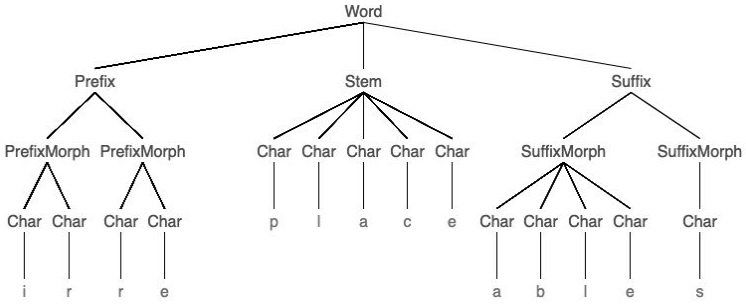
(a) *Morph+SM* grammar



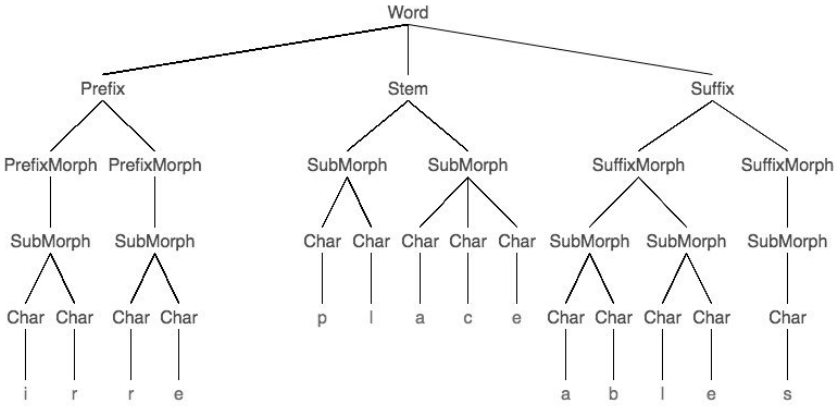
(b) *Simple* grammar



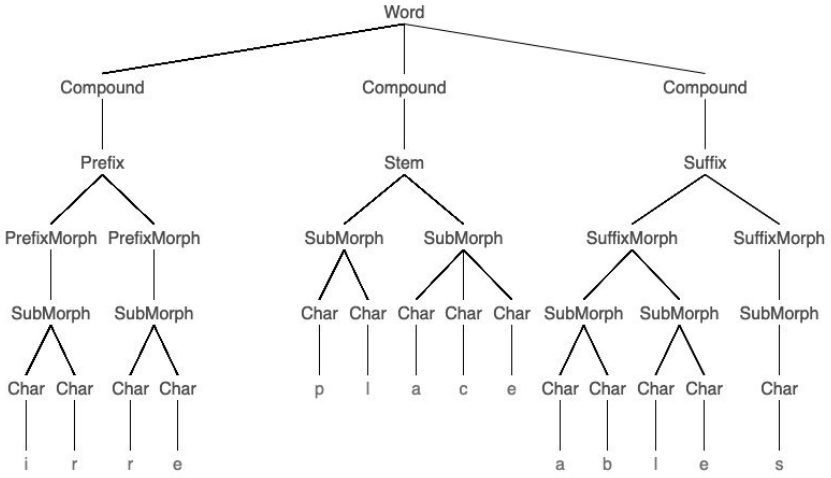
(c) *Simple+SM* grammar



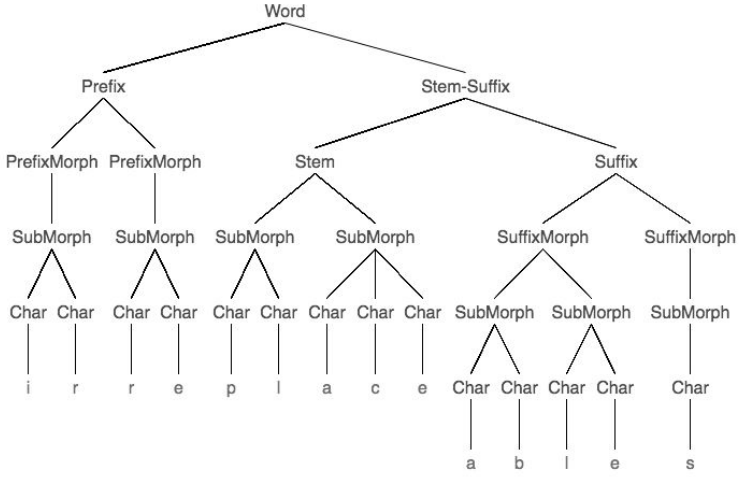
(d) *PrStSu* grammar



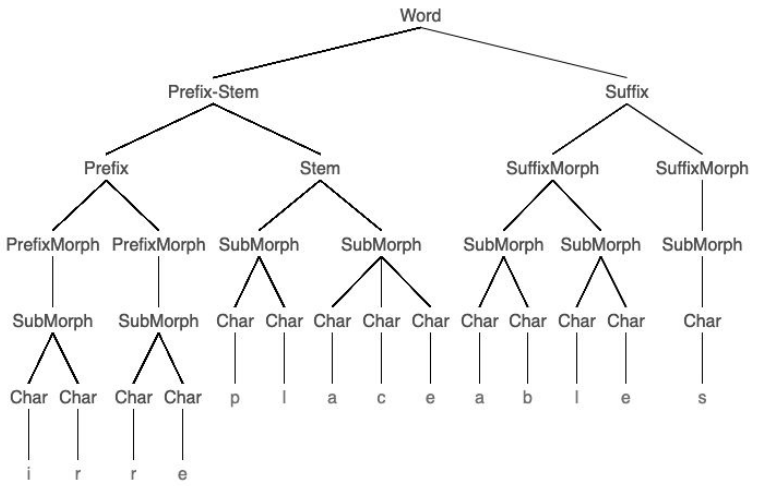
(e) *PrStSu+SM* grammar



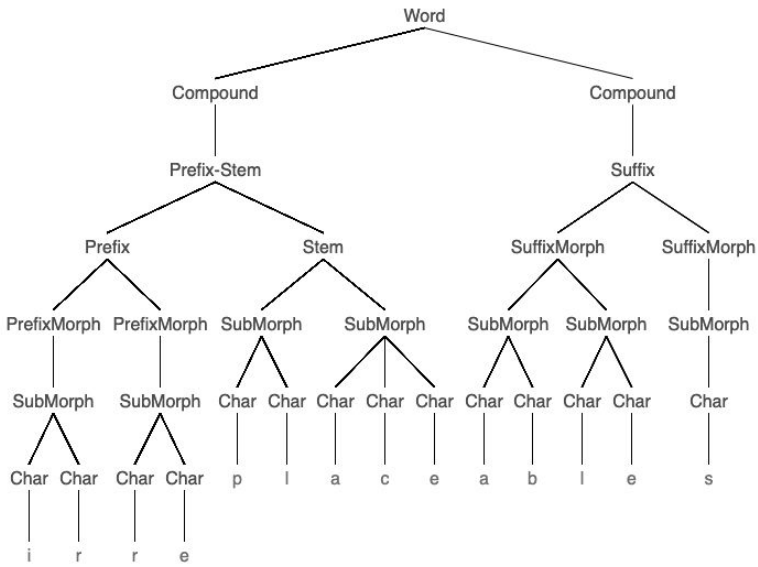
(f) *PrStSu+Co+SM* grammar



(g) *PrStSu2a+SM* grammar



(h) *PrStSu2b+SM* grammar



(i) *PrStSu2b+Co+SM* grammar

Figure 3.1: The representations of the word *irreplaceables* using different grammar definitions

Figure 3.1 shows how the word *irreplaceables* is represented using the nine grammar definitions listed in Table 3.1. However, since *SubMorph* is a nonterminal that is not linguistically driven, its exact instantiation depends on the output of the sampling process given a lexicon and the underlying grammar.

3.3.2 Learning Settings

3.3.2.1 Standard Setting

In this setting, we train a morphological-segmentation model using a language-independent grammar that does not model language-specific characteristics nor contain seeded knowledge about the underlying language. This setting is typically used when processing an unseen language or a language whose description is inadequate or lacking, as in the case of some low-resource and endangered languages. For details on how AGs learn morphological segmentation in a fully unsupervised manner, see Section 3.2.

3.3.2.2 Scholar-Seeded Setting

In this setting, we seed scholar knowledge that is compiled from language resources into the grammars towards a more informed morphological-segmentation model. The intuition behind seeding scholar knowledge is that for many languages, we have more or less extensive descriptions of their morphology, where several online resources provide listings of affixes, usually without contexts, such as the Wiktionary and online grammar references. We therefore investigate the question of whether this data [3] can be used towards a minimally supervised setting, as opposed to a fully unsupervised one.

AGs are a framework that is particularly well suited for applying scholar-seeded knowledge as AGs take as input hand-crafted grammars. Accordingly, we can insert affixes into these grammars in the positions where the affixes are generated, while we continue to allow the grammars to generate

[3] We note that this data is not "data" in the normal sense of machine learning; it is not in the same format as the desired output (i.e., segmented words).

new affixes as we do not expect the scholarly resources to contain complete listings. For these experiments, we consult only online resources, where we spend less than two hours per experimental language to assemble prefixes and suffixes and seed them into the PCFGs as additional production rules.

Figure 3.2 illustrates the *Scholar-Seeded* learning setting, where the prefixes *re*, *im* and *ex* and the suffixes *er* and *s* are seeded into the *PrStSu+SM* grammar as *PrefixMorph* and *SuffixMorph* production rules, respectively, prior to training the morphological-segmentation model.

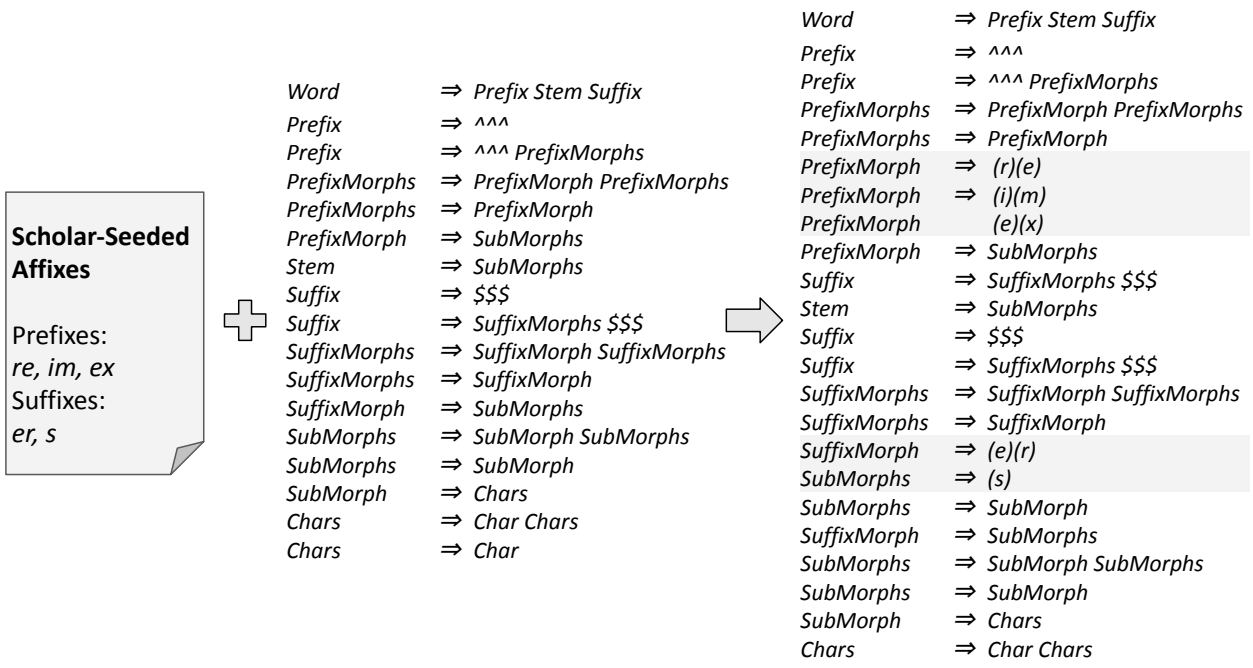


Figure 3.2: An example of the *Scholar-Seeded* setting, where some English affixes, as scholarly knowledge, are seeded into the *PrStSu+SM* grammar

Since the seeded affixes are compiled from language resources, possibly by someone who does not know the underlying language, they are not guaranteed to be correct or highly accurate. Therefore, we seed the affixes into the grammars as unadapted nonterminals as we want to prevent the sampler from spreading wrong information by producing multiple instances of the corresponding subtrees. We verified this hypothesis by seeding the affixes once as adapted and once as unadapted and found that the latter yields better performance across the development languages.

3.3.2.3 Cascaded Setting

The *Cascaded* setting approximates the effect of the *Scholar-Seeded* setting in a language-independent manner through self-training. That is instead of compiling a set of affixes from existing resources, we use affixes obtained from an initial round of learning.

The *Cascaded* setting relies on two rounds of learning. In the first round, we train a morphological-segmentation model using a high-precision grammar in the *Standard* setting and extract the list of the most common affixes from the segmentation output. Next, we seed the list of extracted affixes into the grammar of interest as unadapted nonterminals, in a similar way to how the affixes are seeded in the *Scholar-Seeded* setting, and train another morphological-segmentation model using the augmented grammar in a second round of learning.

We choose a grammar to be the basis for our *Cascaded* setting independently of the language as the aim is to derive language-independent morphological segmentation. We do so by optimizing on precision. The reason to choose high precision (rather than high F1-score) is that we want to be certain of having true affixes in the grammar, rather than having as many affixes as possible (even if some are incorrect). Therefore, we choose the *PrStSu2b+Co+SM* grammar as it achieves the highest on-average precision when evaluated on our development languages (Tables 3.7 and 3.8). In addition, we ran experiments in which we extracted and seeded n affixes, where $n \in \{10, 20, 30, 40, 50, 100\}$, and found that $n = 40$ yields the best on-average performance across the development languages. Accordingly, we set $n = 40$ in all of our *Cascaded* setups.

Figure 3.3 illustrates the *Cascaded* learning setting, where the prefixes *re*, *im* and *ex* and the suffixes *er* and *s* are generated from a first round of learning using the *PrStSu2b+Co+SM* grammar and seeded into the *PrStSu+SM* grammar as *PrefixMorph* and *SuffixMorph* production rules, respectively, for a second round of learning.

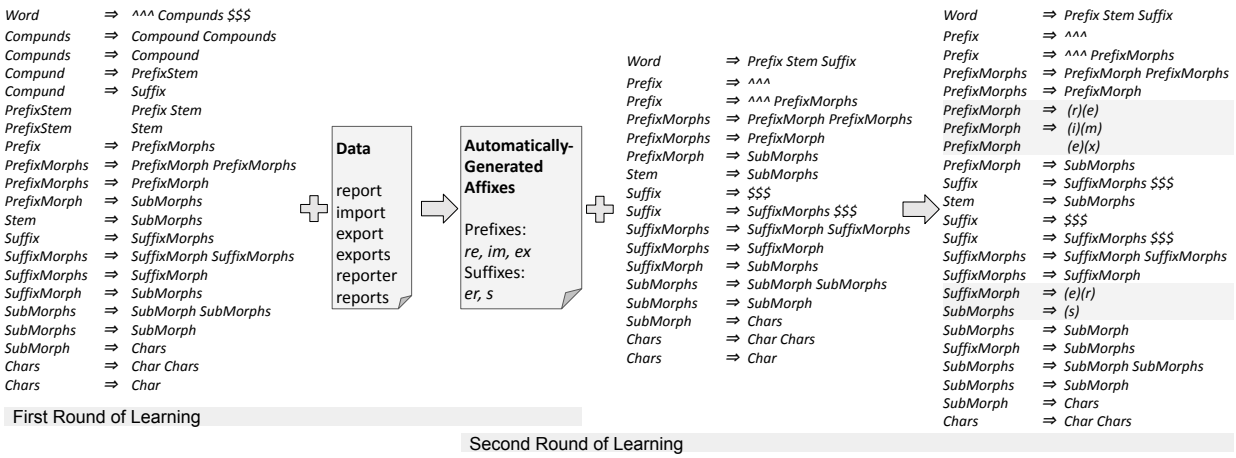


Figure 3.3: An example of the *Cascaded* setting, where some English affixes are extracted from an initial round of learning using the *PrStSu2b+Co+SM* grammar and seeded into the *PrStSu+SM* grammar for a second round of learning

3.3.3 Automatic Tailoring of Grammars for Unseen Languages

3.3.3.1 Picking a Language-Independent Configuration

The experiments on our development languages show that the *PrStSu+SM* grammar results in the best on-average performance of 71.2% in BPR F1-score in the language-independent settings, *Standard* and *Cascaded*, where the *Standard PrStSu+SM* configuration yields the best language-independent performance for English, and German, while the *Cascaded PrStSu+SM* configuration yields the best language-independent performance for Finnish and Turkish (Table 3.7). Accordingly, we choose to use the *PrStSu+SM* grammar to process any unseen language (a language that is not part of the development). However, the preference between the *Standard* and *Cascaded* settings differs across the development languages. We therefore exploit machine learning to derive a method that chooses between the two settings for any given language of unexplored morphology.

We build a binary model that chooses between the *Standard* and *Cascaded* settings given only six data points: two languages with the *Standard* class, namely English and German, and four languages with the *Cascaded* class, namely Finnish, Estonian, Turkish and Zulu.

In order to extract learning features for the classification task, we conduct a learning round

using the *Standard PrStSu+SM* configuration for 50 optimization iterations (one tenth of the number of iterations in a complete sampling phase (Section 3.5.1)) as the purpose is to quickly generate morphological clues that help in the classification rather than to obtain highly optimized morphological segmentation. We choose the *Standard PrStSu+SM* configuration due to its high efficiency across all the development languages in addition to its relatively short sampling time. We then parse the segmentation output to extract 18 morphological features for classification. The features are listed in Table 3.2 [4], where a simple affix contains only one morph, while a complex affix contains one or more simple affixes.

Feature ID	Feature Description
F01	Number of distinct simple prefixes
F02	Average number of simple prefixes per word
F03	Average number of characters per simple prefix
F04	Number of distinct simple suffixes
F05	Average number of simple suffixes per word
F06	Average number of characters per simple suffix
F07	Number of distinct simple affixes
F08	Average number of simple affixes per word
F09	Average number of characters per simple affix
F10	Number of distinct complex prefixes
F11	Average number of complex prefixes per word
F12	Average number of characters per complex prefix
F13	Number of distinct complex suffixes
F14	Average number of complex suffixes per word
F15	Average number of characters per complex suffix
F16	Number of distinct complex affixes
F17	Average number of complex affixes per word
F18	Average number of characters per complex affix

Table 3.2: Classification features for the automatic selection of the language-independent setting

In the training phase, we perform leave-one-out cross-validation on the six development languages, where in each of the six folds of the cross-validation, we choose one language in turn as the test language. We experiment with three classification methods, namely K-Nearest Neighbors, Ad-

[4]We only consider affixes that appear at least 10 times in the segmentation output.

aBoost [5] and Naive Bayes, and report the gold and predicted classes for each language in Table 3.3, where the gold standard is based on the results in Table 3.7. As illustrated, AdaBoost yields the best performance as it predicts the right setting for all the development languages except English, in which the results of the *Standard* and *Cascaded* settings differ by a BPR F1-score of only 0.7%. Accordingly, we choose AdaBoost for classification.

Language	Gold Setting	Classification		
		K-Nearest Neighbors	AdaBoost	Naive Bayes
English	*Standard*			
German	*Standard*	✓	✓	
Finnish	*Cascaded*	✓	✓	✓
Estonian	*Cascaded*		✓	✓
Turkish	*Cascaded*		✓	✓
Zulu	*Cascaded*	✓	✓	✓
	Accuracy %	50.0	83.3	66.7

Table 3.3: The gold and automatically selected language-independent settings per development language

We call our selection approach *AG-LI-Auto*; it works as follows: For an unseen language, we first apply the *Standard PrTuSu+SM* configuration for 50 optimization iterations in order to obtain the values of the morphological features listed in Table 3.2. We then apply the AdaBoost classifier on those features in order to obtain the recommended language-independent setting (*Standard* or *Cascaded*). We finally apply the *PrStSu+SM* grammar in the recommended setting for morphological segmentation.

Studying the correlation between the morphological features and the output of the AdaBoost classifier shows that features F05, F13 and F16, in Table 3.2, namely the average number of simple suffixes per word, the number of distinct complex suffixes and the number of distinct complex affixes, are the most significant ones for the selection of the best setting. This illustrates the high reliance on information about suffixes as the three significant features are suffix-related.

[5]The randomization in AdaBoost gives different outcomes for Turkish, so we ran the training and testing phases for 100 times and voted for the most common outcome for Turkish.

3.3.3.2 Picking a Language-Dependent Configuration

The experiments on our development languages show that the *PrStSu+SM* and *PrStSu2a+SM* grammars result in the best on-average performance of 73.2% and 72.0% in BPR F1-score, respectively, in the *Scholar-Seeded* setting, where the *Scholar-Seeded PrStSu+SM* configuration yields the best *Scholar-Seeded* performance for English, German and Finnish, while the *Scholar-Seeded PrStSu2a+SM* configuration yields the best *Scholar-Seeded* performance for Turkish (Table 3.7). Accordingly, we derive a simple method to choose between the two grammars for the morphological segmentation of any given language for which an evaluation dataset is not accessible, but one can compile a set of affixes to conduct the *Scholar-Seeded* setting.

First, we conduct morphological segmentation using the *Scholar-Seeded PrStSu+SM* and *Scholar-Seeded PrStSu2a+SM* configurations for the underlying language. We then parse the segmentation outputs to extract the most frequent n affixes in each configuration, where we empirically set $n = 100$. Finally, we choose the grammar that results in the highest number of common affixes between the segmentation output and the seeded knowledge. We call this selection approach *AG-SS-Auto*.

Language	Gold Grammar	Automatic Selection
English	*PrStSu+SM*	✓
German	*PrStSu+SM*	✓
Finnish	*PrStSu+SM*	✓
Estonian	*PrStSu2a+SM*	
Turkish	*PrStSu2a+SM*	✓
Zulu	*PrStSu+SM*	✓
	Accuracy %	83.3

Table 3.4: The gold and automatically selected grammars per development language in the *Scholar-Seeded* setting

Table 3.4 reports the gold grammar (from Table 3.7) and the automatically selected one for each development language. Our selection method is able to pick the correct grammar for all the development languages except Estonian, in which the results of the *Scholar-Seeded PrStSu+SM* and *Scholar-Seeded PrStSu2a+SM* configurations differ by an F1-score of only 0.3%.

3.3.4 Incorporating Linguistic Priors

We next propose the use of strong linguistic priors within AGs in order to enhance morphological segmentation in a minimally supervised manner. We introduce two types of priors:

- **Grammar Definition**: A language-specific grammar that models specific morphological phenomena is tailored for the language of interest.
- **Linguist-provided Affixes**: An expert in the underlying language compiles a list of carefully selected affixes and seeds it into the grammars prior to training the morphological-segmentation model.

3.3.4.1 Linguistic Priors as Grammar Definition

While the grammars described in Section 3.3.1 are intended to be generic and to describe word structure in any language, we hypothesize that a definition that imposes language-specific constraints is more efficient. Therefore, we investigate the incorporation of linguistic priors in the form of a grammar definition that models language-specific morphological phenomena. We utilize the best on-average performing grammar *PrStSu+SM* with Japanese as a case study.

The language-independent grammar definition and its Japanese cognate are illustrated in Figure 3.4 on the left and right sides, respectively. We impose the following specifications for Japanese:

1. A word has a maximum of one one-character or two-character prefix.
2. A stem is recursively defined as a sequence of morphs in order to allow for stem compounding.
3. Characters are separated into two groups, Kana (Japanese syllabaries) and Kanji (adopted Chinese characters).
4. *SupMorph* represents a sequence of characters that is either in Kana or Kanji.

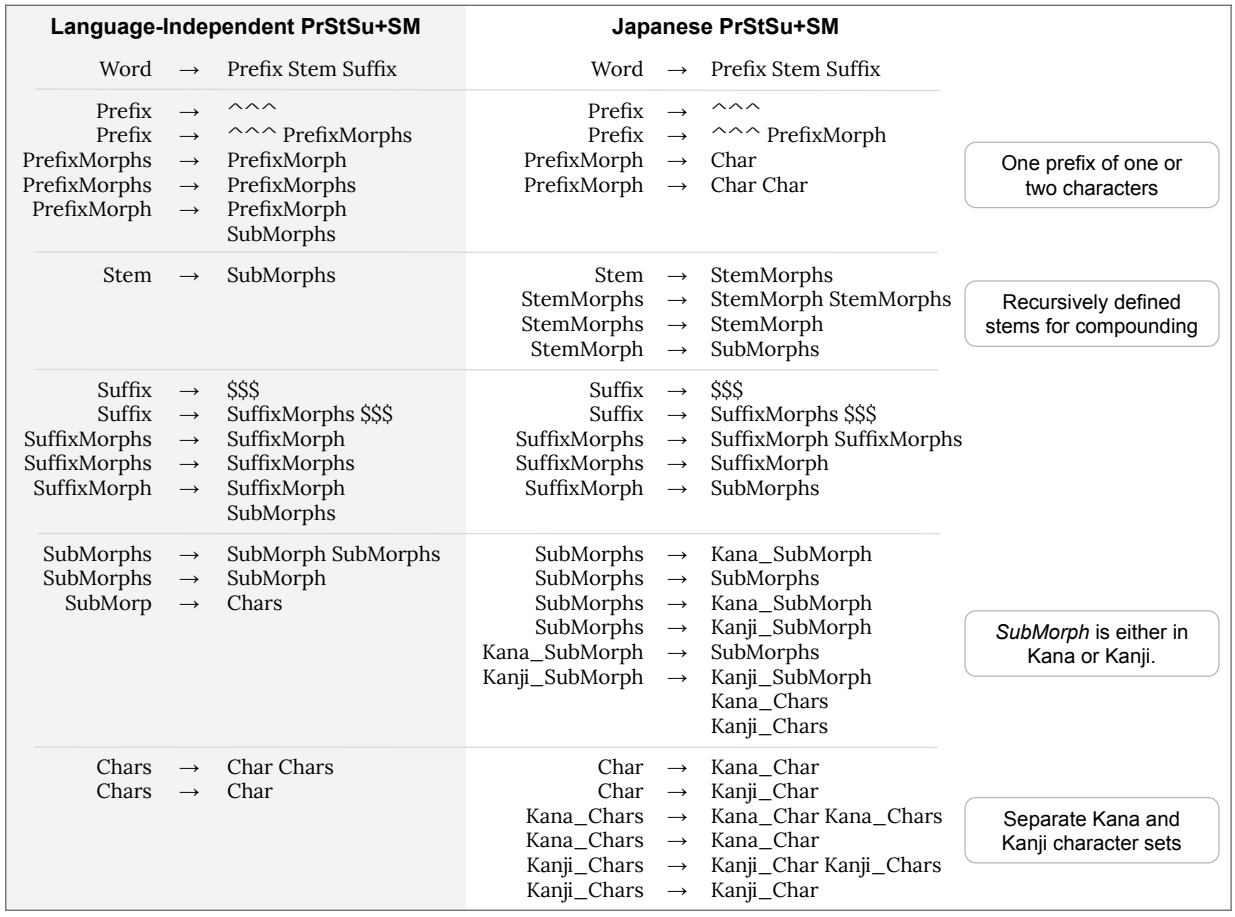


Figure 3.4: The language-independent *PrStSu+SM* grammar (left side) versus its Japanese cognate (right side)

3.3.4.2 Linguistic Priors as Linguist-Provided Affixes

Similar to the *Scholar-Seeded* setting, described in Section 3.3.2.2, we compile a set of affixes and seed it into the grammar trees before training the morphological-segmentation model. However, a major difference is that in the *Scholar-Seeded* setting, the linguistic priors are weak as they are generated from online resources by someone who may have never studied the underlying language, where the purpose is to quickly collect additional clues for the sampler, while here we seed strong priors that are carefully compiled by an expert who specializes in the underlying language. Another difference is that the affixes are seeded in the *Scholar-Seeded* setting as unadapted nonterminals in order to prevent the sampler from spreading wrong information, while here we seed the affixes

as adapted nonterminals since they are guaranteed to be of high quality, and thus instantiating corresponding subtrees in encouraged.

We use Georgian and Arabic as two case studies for the use of linguist-provided affixes. In the case of Georgian, a linguist who is an expert in Georgian, as a second language, compiles a set of 119 affixes [6] that are collected from the leading reference grammar book by Aronson (1990), while in the case of Arabic, a computational linguist who is a native speaker of Arabic compiles a set of 33 affixess [7].

3.3.5 Multilingual Morphological Segmentation

We conduct multilingual training in which we combine lexicons from different languages that are closely related. We examine the case where the combined languages belong to the same language family and share some morphemes, to different degrees, in low-resource scenarios. Our assumption is that shared information across related languages can compensate for lacking information due to limited vocabularies. More specifically, missing morphemes/morphs in the lexicon of one language can be learned from the lexicon of another.

3.4 Languages and Data

We consider 13 languages that are spread across the typology spectrum and for which morphologically segmented datasets are available for evaluation. Six out of the 13 languages are development ones that we use to derive the main conclusions concerning our grammar definitions, learning settings and the automatic tailoring of grammars for unseen languages. These languages are English, German, Finnish, Estonian, Turkish and Zulu. The other languages are test ones, namely Japanese, Georgian, Arabic, Mexicanero, Nahuatl (Mexicano), Wixarika (Huichol) and Mayo (Yorem Nokki). Information about the languages and their datasets are listed in Table 3.5.

[6] https://github.com/rnd2110/MorphAGram/blob/master/data/georgian/data/elk.txt
[7] https://github.com/rnd2110/MorphAGram/blob/master/data/arabic/data/elk.txt

Language	Typology	Source	Number of Words		
			TRAIN	*DEV*	*TEST*
English	Analytic	Morpho Challenge	50,000	1,212	NA
German	Fusional, more Synthetic	Morpho Challenge	50,000	556	NA
Finnish	Agglutinative, more Synthetic	Morpho Challenge	50,000	1,494	NA
Estonian	Agglutinative, more Synthetic	Sega Corpus	49,621	1,492	NA
Turkish	Agglutinative, more Synthetic	Morpho Challenge	50,000	1,531	NA
Zulu	Agglutinative, mildly Fusional	Ukwabelana Corpus	50,000	1,000	NA
Japanese	Agglutinative, more Synthetic	Wikipedia	48,423	NA	1,000
Georgian	Agglutinative, more Polysynthetic	Wikipedia	50,000	NA	1,000
Arabic	Fusional, less Synthetic	PATB	50,000	NA	1,000
Mexicanero	Polysynthetic	Kann et al. (2018)	424	106	351
Nahuatl	Polysynthetic	Kann et al. (2018)	535	133	439
Wixarika	Polysynthetic	Kann et al. (2018)	664	166	546
Mayo	Polysynthetic	Kann et al. (2018)	509	126	419

Table 3.5: Typological and data-related information per experimental language. NA = Not applicable.

English, German, Finnish and Turkish The data is compiled from the Morpho Challenge competition [8] (MC2010) (Kurimo et al., 2010), where we select the most frequent 50,000 words for training after filtering out the words that contain foreign letters. In addition, the gold-standard development sets are collected from all the years of the competition.

Estonian The training and gold-standard development sets are the ones used by Sirts and Goldwater (2013) [9] after filtering out the words that contain foreign letters. The data is based on the Sega corpus [10], where the gold segmentation is collected from the Estonian Morphologically Disambiguated Corpus [11].

Zulu The training data is collected from the Ukwabelana corpus (Spiegler et al., 2010), an open-source Zulu corpus that is morphologically annotated, while the words in the gold-standard development set are a randomly selected subset.

[8] http://research.ics.aalto.fi/events/morphochallenge2010/datasets.shtml
[9] through contacting the authors directly
[10] https://keeleressursid.ee/et/196-segakorpus-eesti-ekspress
[11] https://www.cl.ut.ee/korpused/morfkorpus

Japanese The training data is based on the most frequent 48,423 words in the Japanese Wikipedia, while the gold-standard test set is based on 1,000 randomly selected words from the training set [12] by contracting a native-speaker linguist for the annotation of the gold segmentation.

Georgian The training data is based on the most frequent 50,000 words in the Georgian Wikipedia, while the gold-standard test set is in-house annotations conducted for 1,000 randomly selected words from the training set [13]. The annotations were first prepared by a non-linguist who is a native Georgian speaker. A linguist who speaks Georgian as a second language then revised the annotations in a second phase, corrected 193 segmentation examples and further annotated 116 words for additional alternative segmentation. We verified the fact that having the gold annotations revised and corrected by a linguist improves evaluation quality.

Table 3.6 shows examples of the manually annotated Japanese and Georgian gold segmentation. In the case of Georgian, in addition to morphological segmentation, the linguist coded each word based on its syntactic category: nominal (475 words), verbal (359 words), numeral (44 words) and other (122 words).

Japanese Word	Segmentation
いました	い + ま + した
勉強して	勉強 + し + て
始められません	始め + られ + ま + せん

Georgian Word	Category	Segmentation
თვითთვრინავი	Nominal	თვი + თ + თვრინ + ავ + ი თვითთვრინავ + ი
იქნებაო	Verbal	ი + ქნ + ებ + ა + ო იქნებ + ა + ო
თოთხმეთი	Numeral	თ + ოთხ + მეთ + ი თოთხმეთ + ი
ვიხი	Other	ვი + ხ + ი ვიხ + ი

Table 3.6: Japanese and Georgian segmentation examples

[12] https://github.com/rnd2110/MorphAGram/blob/master/data/japanese/data/japanese.dev.gold
[13] https://github.com/rnd2110/MorphAGram/blob/master/data/georgian/data/georgian.dev.gold

Arabic The training data is collected from the most frequent 50,000 words in the third release of the Penn Arabic Treebank (PATB) (Maamourio et al., 2004) after removing diacritization, while the gold-standard test set is based on a randomly selected subset of 1,000 words.

Mexicanero, Nahuatl, Wixarika and Mayo We use the datasets released by Kann et al. (2018) after filtering out the words that are not white-space tokenized or containing foreign letters. The four languages are polysynthetic ones and come in small datasets of less than 1,000 words, which is challenging for the task of morphological segmentation in terms of morphological complexity and data availability.

In all the languages, we train our models using the training sets (*TRAIN*) without seeing gold-standard segmentation. For evaluation and analysis, we use the development sets (*DEV*) of the development languages and the test sets (*TEST*) of Japanese, Georgian and Arabic. In the case of the polysynthetic languages, we report the results on the development sets (*DEV*) released by Kann et al. (2018), unless otherwise noted.

In the case of German, the gold segmentation includes actual morphemes as opposed to their inflected forms (morphs). For example, the word *wohlgefällige* is segmented as *wohl+ge+fall+ig+e* instead of *wohl+ge+fäll+ig+e*. However, since our system produces morph-based segmentation, we processed the gold examples in which the concatenation of the morphemes does not yield the surface word, where we tried to generate the gold morphs using simple rules that replace the vowels in the morphemes by their corresponding umlauts before segmentation. We then excluded all the gold examples in which the generated morphs do not form the surface word, which eliminated 29.2% of the gold examples. We however report results when evaluating on the original gold-standard development set, in addition to the filtered one, for completeness.

For the *Scholar-Seeded* affixes, we mainly rely on the Wiktionary to collect prefixes and suffixes of the language of interest. If the Wiktionary does not provide a sufficient number of affixes, we collect additional ones from grammar pages that we obtain by searching the Web. We however keep the process of collecting the set of affixes less than two hours in length per language in order to

preserve our low-resource settings.

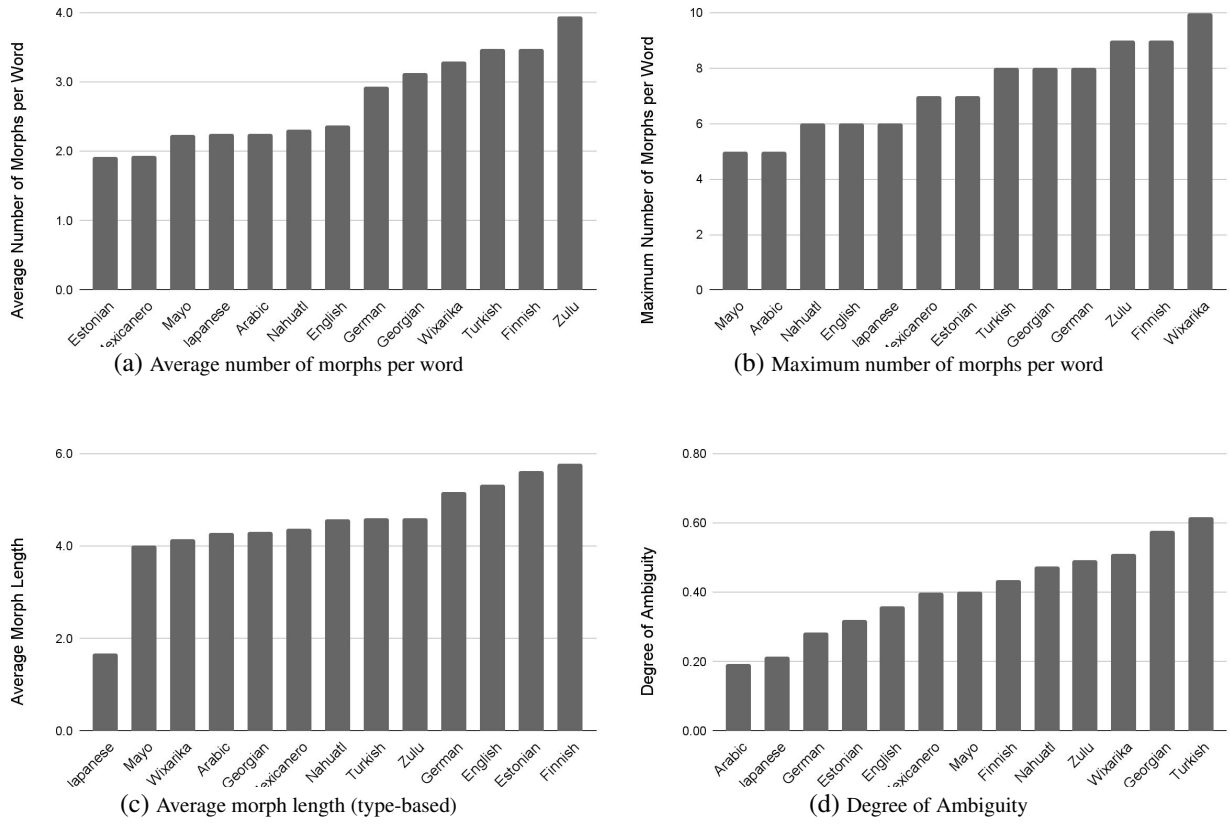
We define what we call the degree of ambiguity. Assume we have a total of n morphs, where $i \in [1, n]$ indexes a morph, N_{i} is the number of occurrences of morph i and M_{i} is the number of occurrences of the surface form constituting morph i, then the degree of ambiguity is defined as:

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

The degree of ambiguity indicates how ambiguous the morphs in the language are. A morph is unambiguous if its surface form either unlikely or most likely represents the morph. For instance, the English suffix *ly* is unambiguous as *ly* is unlikely part of a stem, and the English prefix *ab* is unambiguous as *ab* is most likely part of a stem. In contrast, the common verbal suffix *t* in German is ambiguous since an ending *t* is part of several stems. The division $\frac{N_{\mathrm{i}}}{M_{\mathrm{i}}}$ in the formula above represents the probability that a specific surface form of a morph represents the morph, where a value near 0.5 indicates that the morph is highly ambiguous, while summing and dividing over n is for averaging across the morphs in the underlying language.

Figure 3.5 reports morph-level statistics for our experimental languages. This includes the average number of morphs per word (Figure 3.5a), the maximum number of morphs per word (Figure 3.5b), the average morph length (type-based) (Figure 3.5c) and the degree of ambiguity (Figure 3.5d).

Zulu, Finnish and Turkish have the highest average number of morphs per word of 3.9, 3.5 and 3.5 respectively. We therefore expect low morphological-segmentation recall for these three languages. Turkish is getting further interesting for the task of morphological segmentation as it has the highest degree of ambiguity of 0.6. On another hand, Wixarika witnesses the maximum number of 10 morphs per word, with an average of 3.3 morphs per word, lending itself as a strong polysynthetic language.



(a) Average number of morphs per word

(b) Maximum number of morphs per word

(c) Average morph length (type-based)

(d) Degree of Ambiguity

Figure 3.5: Morphological statistics

3.5 Evaluation and Analysis

3.5.1 Experimental Settings

We conduct our experiments in a transductive learning scenario, where the unsegmented words in the evaluation set are included in the training set, which is common in the evaluation of unsupervised morphological segmentation (Poon et al., 2009; Sirts and Goldwater, 2013; Narasimhan et al., 2015). However, we do not see significant performance drops when adopting the inductive approach instead, in which the training and evaluation sets do not include examples in common.

For training, we run the sampler for 500 optimization iterations for all the languages [14]. In addition, no annealing is used as it does not improve the results, and all the hyperparameters of the

[14] In a few cases, we run the sampler for fewer iterations in order to speed up the segmentation process. This includes the execution of the *PrStSu2b+Co+SM* grammar in the cases of Turkish, Zulu, Georgian and Arabic, the seeded *PrStSu2b+SM* grammar for Zulu and the seeded *PrStSu+Co+SM* grammar for Japanese

model are automatically inferred. For evaluation, we compute all the results in this chapter as the average of five runs since the sampler is non-deterministic.

We evaluate the performance of our morphological-segmentation framework *MorphAGram* using two metrics: Boundary Precision and Recall (BPR) and EMMA-2 (Virpioja et al., 2011). BPR is the classical evaluation method for morphological segmentation. It measures the ability of the system to detect segmentation boundaries, where the boundaries in the proposed segmentation are compared to the boundaries in the reference. On the other hand, EMMA-2 measures the ability of the system to detect the morphemes/morphs. EMMA-2 is a variation of EMMA (Spiegler and Monson, 2010), in which each proposed morpheme/morph is matched to each morpheme/morph 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/morphs can be mapped to one reference morpheme/morph. EMMA-2 usually results in higher precision and recall than EMMA and BPR as it tolerates failing to join two allomorphs or to distinguish between identical syncretic morphemes/morphs. For more details about the evaluation metrics, see Virpioja et al. (2011).

We evaluate our system versus two state-of-the-art baselines: *Morfessor* (Creutz and Lagus, 2007; Grönroos et al., 2014) and *MorphoChain* (Narasimhan et al., 2014). *Morfessor* is a commonly used framework for unsupervised and semi-supervised morphological segmentation and is publicly available [15]. *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 is also based on an HMM that encodes the positional information of the morphs. *MorphoChain* is another publicly available system for unsupervised morphological segmentation [16]. In *MorphoChain*, words are modeled as morphological chains, where a chain is a sequence of words that starts with a base word (a 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, along with correlation and similarity measurements, to derive the morphological segmentation.

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

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

3.5.2 Performance of All Grammars

Tables 3.7 and 3.8 report the performance of our *MorphAGram* framework using the BPR and EMMA-2 metrics, respectively, for the development languages (English, German, Finnish, Estonian, Turkish and Zulu) using the nine grammars defined in Section 3.3.4.2.

Considering the BPR metric, there is a vast variation among the languages in how the grammars perform. For instance, in the *Standard* Setting, the *PrStSu+SM* grammar yields the best F1-score for English, German and Turkish, while the *PrStSu2a+SM*, *PrStSu+Co+SM* and *PrStSu2b+SM* grammars yield the best F1-score for Finnish, Estonian and Zulu, respectively. We get a similar pattern in the *Cascaded* setting except in the cases of Finnish and Zulu, in which the *PrStSu+SM* and *Simple* grammars yield the best F1-score, respectively. However, the best grammar in the *Scholar-Seeded* setting is the same as the one in the *Cascaded* setting except in the cases of Turkish and Zulu, in which the *PrStSu2a+SM* and *PrStSu+SM* grammars yield the best F1-score, respectively.

When averaging across the development languages, the *PrStSu+SM* grammar gives the best on-average F1-score in the *Standard*, *Cascaded* and *Scholar-Seeded* settings, achieving F1-scores of 68.9%, 73.5% and 73.2%, respectively. The *PrStSu2a+SM* grammar then comes second in the three settings with F1-scores of 68.6%, 69.0% and 72.0%, respectively. In contrast, the *PrStSu2b+Co+SM* grammar gives the lowest F1-score in the three settings, but it consistently yields the highest precision, which makes it ideal for the first round of learning in the *Cascaded* setting, where the purpose is to use a conservative grammar that produces true affixes that we can confidently seed in the second learning round. On the other hand, the *PrStSu* grammar achieves the highest recall in the three learning settings but at the cost of relatively low precision.

Considering the EMMA-2 metric, we experience similar patterns to those of the BPR metric. The only differences are that 1) the *Morph+SM* grammar yields the best F1-score for Estonian in the *Cascaded* setting instead of *PrStSu+Co+SM*; 2) the *PrStSu* grammar gives the best F1-score for Zulu in the *Cascaded* setting instead of *Simple*; and 3) the *PrStSu2a+SM* grammar yields the best on-average F1-score in the *Standard* and *Scholar-Seeded* settings instead of *PrStSu+SM*. However, none of these differences is statistically significant for $p\text{-}value < 0.01$.

Language	Grammar	*Standard* Setting			*Cascaded* Setting			*Scholar-Seeded* Setting		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
English	*Morph+SM*	79.6	69.8	74.3	80.2	70.0	**74.8**	79.1	69.0	73.7
	Simple	50.7	64.7	56.9	51.6	64.7	57.4	51.1	65.1	57.2
	Simple+SM	70.4	63.7	66.9	70.1	63.9	66.9	69.6	63.5	66.4
	PrStSu	41.2	79.7	54.3	56.7	82.6	67.3	64.6	83.2	72.7
	PrStSu+SM	72.7	78.6	**75.5**	72.0	77.9	**74.8**	75.3	78.5	**76.9**
	PrStSu+Co+SM	86.1	66.3	74.9	86.5	65.1	74.3	85.6	66.6	74.9
	PrStSu2a+SM	67.5	74.9	70.9	72.8	70.8	71.8	74.2	70.3	72.2
	PrStSu2b+SM	48.1	71.4	57.5	49.8	74.6	59.7	50.6	75.5	60.6
	PrStSu2b+Co+SM	98.3	22.7	36.9	98.2	23.2	37.5	98.2	22.7	36.8
German	*Morph+SM*	86.6	62.7	72.7	86.1	62.0	72.1	86.9	62.2	72.5
	Simple	67.9	63.5	65.6	67.9	63.5	65.6	68.5	64.1	66.2
	Simple+SM	83.6	65.0	73.1	83.4	64.9	73.0	83.0	64.8	72.7
	PrStSu	59.7	83.9	69.7	66.6	84.2	74.4	70.1	86.4	77.4
	PrStSu+SM	81.7	74.9	**78.1**	81.4	74.7	**77.9**	81.3	76.0	**78.6**
	PrStSu+Co+SM	91.1	55.3	68.8	90.2	57.6	70.3	90.5	57.3	70.2
	PrStSu2a+SM	81.7	71.5	76.3	79.5	69.4	74.1	80.1	70.1	74.7
	PrStSu2b+SM	76.0	75.4	75.7	74.7	75.3	75.0	74.5	74.9	74.7
	PrStSu2b+Co+SM	97.0	20.1	33.2	96.8	24.1	38.6	96.7	23.3	37.5
Finnish	*Morph+SM*	77.0	53.0	62.8	76.9	53.1	62.8	77.0	53.5	63.1
	Simple	59.8	51.8	55.6	60.7	52.5	56.3	60.7	52.3	56.2
	Simple+SM	76.0	49.1	59.6	76.0	49.1	59.7	75.7	48.7	59.3
	PrStSu	46.5	59.8	52.3	56.7	72.3	63.6	50.4	76.3	60.7
	PrStSu+SM	63.2	54.7	58.6	77.7	66.4	**71.6**	77.5	68.6	**72.8**
	PrStSu+Co+SM	83.2	50.0	62.5	84.0	51.0	63.5	82.9	51.7	63.7
	PrStSu2a+SM	71.5	67.1	**69.2**	78.0	61.5	68.8	72.0	67.4	69.6
	PrStSu2b+SM	60.5	56.6	58.5	62.8	59.6	61.2	62.3	59.6	60.9
	PrStSu2b+Co+SM	96.9	22.5	36.6	96.7	23.9	38.4	97.0	22.0	35.9
Estonian	*Morph+SM*	72.9	77.6	75.2	73.1	77.5	75.2	73.4	77.7	75.5
	Simple	47.3	73.8	57.6	46.1	72.4	56.3	46.8	73.7	57.3
	Simple+SM	65.2	75.9	70.2	65.8	76.6	70.8	65.5	76.2	70.4
	PrStSu	38.1	84.2	52.5	53.5	87.5	66.4	49.4	79.8	61.0
	PrStSu+SM	54.6	83.6	65.9	65.9	84.1	73.9	63.9	82.5	72.0
	PrStSu+Co+SM	78.1	76.6	**77.4**	78.1	72.9	**75.4**	78.1	76.8	**77.4**
	PrStSu2a+SM	53.4	86.0	65.9	66.6	78.6	72.1	67.3	78.0	72.3
	PrStSu2b+SM	44.5	84.1	58.2	45.5	84.2	59.1	45.6	84.3	59.2
	PrStSu2b+Co+SM	98.8	28.5	44.3	98.8	28.8	44.6	98.8	29.5	45.4
Turkish	*Morph+SM*	87.1	54.1	66.7	87.3	54.4	67.0	87.6	54.3	67.1
	Simple	71.6	57.1	63.5	71.6	56.7	63.3	71.6	56.8	63.3
	Simple+SM	88.1	51.5	65.0	88.6	51.8	65.4	88.4	51.7	65.2
	PrStSu	58.9	70.7	64.3	72.4	76.7	74.5	60.8	72.3	66.1
	PrStSu+SM	87.5	71.8	**78.9**	88.7	72.5	**79.8**	69.8	60.8	65.0
	PrStSu+Co+SM	89.5	47.5	62.1	89.8	50.7	64.8	89.0	50.8	64.7
	PrStSu2a+SM	83.9	71.5	77.2	88.4	64.0	74.2	83.7	71.0	**76.8**
	PrStSu2b+SM	63.4	64.1	63.7	66.2	67.1	66.6	66.0	66.2	66.1
	PrStSu2b+Co+SM	99.3	4.9	9.3	98.8	5.1	9.6	97.8	6.3	11.9
Zulu	*Morph+SM*	90.0	36.8	52.2	89.6	36.8	52.1	89.6	36.5	51.9
	Simple	74.8	55.3	63.6	74.4	54.9	**63.2**	74.9	55.4	63.7
	Simple+SM	88.6	38.8	54.0	89.0	39.0	54.3	89.3	38.8	54.1
	PrStSu	59.8	50.4	54.7	75.8	53.0	62.4	74.9	68.3	71.5
	PrStSu+SM	74.3	45.1	56.1	91.2	47.8	62.7	85.0	65.3	**73.8**
	PrStSu+Co+SM	91.9	33.8	49.5	92.1	34.5	50.2	92.3	34.8	50.5
	PrStSu2a+SM	62.9	44.6	52.2	64.0	45.4	53.1	76.9	58.4	66.4
	PrStSu2b+SM	90.4	53.3	**67.0**	89.2	41.0	56.5	87.1	56.6	68.6
	PrStSu2b+Co+SM	99.9	3.2	6.1	99.9	3.2	6.1	98.1	4.8	9.1

Table 3.7: The segmentation performance (BPR) of the different grammars on the development languages. The best result per language-setting pair is in **bold**. The best language-independent result per language is underlined.

Language	Grammar	*Standard* Setting			*Cascaded* Setting			*Scholar-Seeded* Setting		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
English	*Morph+SM*	93.2	78.1	85.0	93.3	78.3	85.2	93.1	77.5	84.6
	Simple	81.4	72.8	76.9	82.2	72.9	77.3	81.6	73.0	77.1
	Simple+SM	92.4	73.1	81.6	92.5	73.5	81.9	92.2	73.2	81.6
	PrStSu	62.3	84.0	71.5	74.2	86.3	79.8	79.3	86.6	82.8
	PrStSu+SM	87.7	85.0	**86.3**	86.8	84.5	**85.6**	88.5	85.1	**86.7**
	PrStSu+Co+SM	95.6	77.0	85.3	96.2	76.0	84.9	95.2	76.8	85.0
	PrStSu2a+SM	87.2	80.3	83.6	90.3	78.7	84.1	90.9	77.9	83.9
	PrStSu2b+SM	74.8	77.4	76.1	75.4	80.7	77.9	75.3	80.9	78.0
	PrStSu2b+Co+SM	100.0	48.8	65.6	100.0	48.8	65.6	100.0	48.7	65.5
German	*Morph+SM*	94.8	67.9	79.1	95.0	68.1	79.4	95.1	68.0	79.3
	Simple	91.8	69.6	79.2	91.6	69.5	79.1	91.5	69.6	79.0
	Simple+SM	95.9	70.6	81.3	95.8	70.5	81.2	95.7	70.5	81.2
	PrStSu	75.5	84.9	79.9	79.2	86.1	82.5	81.5	88.5	84.8
	PrStSu+SM	91.2	78.9	**84.6**	90.9	79.2	**84.7**	90.2	80.4	**85.0**
	PrStSu+Co+SM	96.6	64.0	77.0	96.4	65.5	78.0	96.2	65.2	77.7
	PrStSu2a+SM	93.9	75.4	83.6	93.3	74.8	83.0	93.4	75.2	83.3
	PrStSu2b+SM	92.2	77.5	84.2	91.4	78.0	84.2	90.9	77.9	83.9
	PrStSu2b+Co+SM	99.8	40.7	57.8	99.7	41.7	58.8	99.6	41.5	58.6
Finnish	*Morph+SM*	92.0	58.9	71.8	92.1	58.7	71.7	91.8	59.2	72.0
	Simple	87.4	60.4	71.4	87.5	60.5	71.6	87.7	60.7	71.7
	Simple+SM	94.9	56.4	70.7	94.9	56.4	70.8	94.9	56.4	70.7
	PrStSu	67.1	64.0	65.5	67.7	72.4	70.0	63.6	77.1	69.7
	PrStSu+SM	84.8	59.6	70.0	88.0	69.5	**77.6**	87.3	71.0	**78.3**
	PrStSu+Co+SM	94.5	57.2	71.3	94.6	57.9	71.8	94.2	58.4	72.1
	PrStSu2a+SM	86.1	70.3	**77.4**	92.3	66.6	77.4	86.4	70.5	77.6
	PrStSu2b+SM	80.6	60.6	69.2	81.7	63.3	71.3	81.7	63.5	71.4
	PrStSu2b+Co+SM	99.8	42.4	59.5	99.7	42.7	59.8	99.7	42.2	59.3
Estonian	*Morph+SM*	87.5	86.1	86.8	87.6	86.4	**87.0**	87.6	86.1	86.9
	Simple	74.6	84.2	79.1	74.2	83.5	78.6	74.3	83.9	78.8
	Simple+SM	87.2	85.5	86.4	87.3	85.9	86.6	87.2	85.6	86.4
	PrStSu	53.7	88.4	66.8	66.5	91.1	76.9	68.2	87.0	76.4
	PrStSu+SM	73.2	88.9	80.3	79.7	90.8	84.9	78.4	90.2	83.9
	PrStSu+Co+SM	88.9	86.1	**87.5**	89.6	83.9	86.6	89.2	86.0	**87.6**
	PrStSu2a+SM	71.2	89.5	79.3	86.3	87.2	86.8	86.7	86.7	86.7
	PrStSu2b+SM	63.2	89.7	74.2	63.9	89.6	74.6	64.2	89.7	74.9
	PrStSu2b+Co+SM	99.9	59.2	74.3	99.9	59.2	74.4	99.9	59.4	74.5
Turkish	*Morph+SM*	95.2	42.6	58.9	95.1	42.7	58.9	95.1	42.7	58.9
	Simple	88.7	45.5	60.1	89.4	45.5	60.3	89.2	45.5	60.3
	Simple+SM	97.1	40.3	57.0	97.2	40.5	57.2	97.1	40.5	57.2
	PrStSu	68.7	54.5	60.8	78.0	59.4	67.5	69.8	54.9	61.5
	PrStSu+SM	92.4	55.5	**69.3**	92.8	56.0	**69.9**	82.1	46.0	58.9
	PrStSu+Co+SM	96.3	40.3	56.8	96.1	41.2	57.7	96.0	41.2	57.6
	PrStSu2a+SM	91.1	55.0	68.6	93.8	48.3	63.7	91.0	54.8	**68.4**
	PrStSu2b+SM	76.9	47.4	58.7	78.4	49.3	60.5	78.4	48.7	60.1
	PrStSu2b+Co+SM	100.0	26.0	41.2	100.0	25.9	41.2	100.0	26.2	41.5
Zulu	*Morph+SM*	95.5	38.2	54.6	95.2	38.3	54.7	95.4	38.1	54.5
	Simple	86.5	55.8	67.8	85.9	55.9	67.7	86.0	56.0	67.8
	Simple+SM	95.5	39.0	55.4	95.6	39.4	55.8	95.6	39.1	55.5
	PrStSu	74.8	53.7	62.5	83.4	57.2	**67.8**	81.1	67.9	73.9
	PrStSu+SM	87.9	45.2	59.7	93.4	50.4	65.5	88.9	66.4	**76.0**
	PrStSu+Co+SM	96.2	36.8	53.2	96.2	37.1	53.6	96.2	37.2	53.6
	PrStSu2a+SM	82.9	47.4	60.3	83.5	47.9	60.9	86.7	57.3	69.0
	PrStSu2b+SM	93.0	55.9	**69.8**	94.6	41.6	57.7	92.0	57.4	70.7
	PrStSu2b+Co+SM	100.0	19.4	32.5	100.0	19.4	32.5	99.9	19.9	33.1

Table 3.8: The segmentation performance (EMMA-2) of the different grammars on the development languages. The best result per language-setting pair is in **bold**. The best language-independent result per language is underlined.

One interesting phenomena is that the average number of morphs per word and recall are highly correlated. The correlation is illustrated in Figure 3.6 based on the average BPR recall across all the grammars in the *Standard* setting. The plot starts with Estonian, with the highest on-average recall of 74.5% and the minimum of 1.9 average number of morphs per word, and ends with Zulu, with the lowest on-average recall of 40.1% and the maximum of 3.9 average number of morphs per word. The intuition is that with more morphs per word, the models are more likely to miss segmentation points, and thus recall drops.

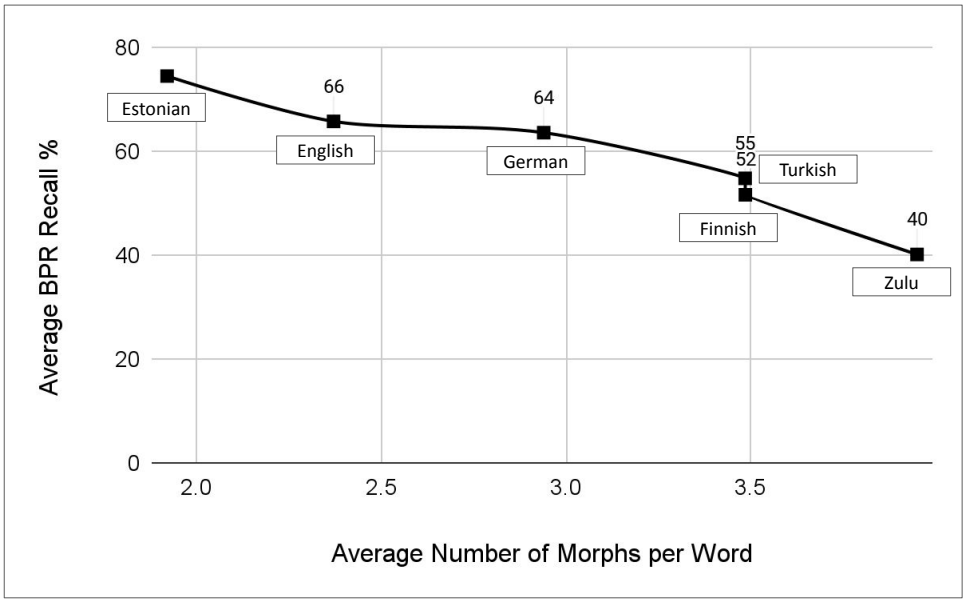


Figure 3.6: Average number of morphs per word versus recall. The calculations are based on the average BPR recall across the grammars in the *Standard* setting.

When we use the original German gold standard, i.e., without filtering (Section 3.4), the performance drops by absolute F1-scores of 35.9%, 36.4% and 36.4% in the *Standard*, *Cascaded* and *Scholar-Seeded* settings, respectively, when using the BPR metric, and by corresponding absolute F1-scores of 13.4%, 14.3% and 13.7%, respectively, when using the EMMA-2 metric. The significant drop in the performance is due to the fact that *MorphAGram* segments words into inflected morphs and does not restore the actual morphemes, and so do the baselines we compare to in the rest of this chapter.

For the performance of the nine grammars on the test languages using the BPR and EMMA-2

metrics, see Tables 1.1 and 1.2, respectively, in Appendix A. The results show similar patterns to those of the development languages, where the *PrStSu+SM* grammar gives the best on-average F1-score in the *Standard*, *Cascaded* and *Scholar-Seeded* settings, followed by the *PrStSu2a+SM* grammar, while the *PrStSu2b+Co+SM* grammar gives the lowest F1-score in the three settings, but it consistently yields the highest precision.

3.5.3 Automatically Selected Configurations versus Upper Bounds

We next apply our approaches for the automatic tailoring of the grammars (Section 3.3.3) to all of our 13 experimental languages. These are 1) *AG-LI-Auto*: to automatically select a language-independent setting (*Standard* or *Cascaded*) for the best on-average *PrStSu+SM* grammar; and 2) *AG-SS-Auto*: to automatically select a grammar (*PrStSu+SM* or *PrStSu2a+SM*) for the *Scholar-Seeded* setting. However, when we apply the *AG-LI-Auto* setup for one of our six development languages, we exclude its corresponding data point from the model. We also compare the performance to *AG-LI-Best* and *AG-SS-Best*, two oracle setups that observe all of our results and choose the best language-independent (*Standard* or *Cascaded*) and *Scholar-Seeded* configurations, respectively. We report the performance in Tables 3.9 and 3.10 using the BPR and EMMA-2 metrics, respectively.

Considering the BPR metric, *AG-LI-Auto* can successfully pick the oracle language-independent configurations of eight languages, namely German, Finnish, Turkish, Japanese, Arabic, Mexicanero, Nahuatl and Wixarika. In the other five languages, *AG-LI-Auto* picks configurations that result in an average F1-score drop of only 2.5% as compared to the oracle configurations. On the other hand, *AG-SS-Auto* can successfully pick the oracle *Scholar-Seeded* configurations of nine languages, namely English, German, Finnish, Turkish, Zulu, Arabic, Mexicanero, Nahuatl and Wixarika. In the other four languages, *AG-SS-Auto* picks configurations that result in an average F1-score drop of only 2.3% as compared to the oracle configurations.

Considering the EMMA-2 metric, we observe similar patterns to those of BPR except that *AG-LI-Auto* fails to pick the oracle language-independent configuration of German, while *AG-SS-Auto* is able to pick the oracle *Scholar-Seeded* configuration of Japanese. In addition, *AG-LI-Auto*

Language	Setup	Setting	Grammar	Prec.	Recall	F1
English	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	72.0	77.9	74.8
	AG-LI-Best	*Standard*	*PrStSu+SM*	72.7	78.6	75.5
German	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	81.7	74.9	78.1
Finnish	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	77.7	66.4	71.6
Estonian	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	65.9	84.1	73.9
	AG-LI-Best	*Standard*	*PrStSu+Co+SM*	78.1	76.6	77.4
Turkish	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	88.7	72.5	79.8
Zulu	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	91.2	47.8	62.7
	AG-LI-Best	*Standard*	*PrStSu2b+SM*	90.4	53.3	67.0
Japanese	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	81.5	78.2	79.8
Georgian	*AG-LI-Auto*	*Standard*	*PrStSu+SM*	82.0	69.1	75.0
	AG-LI-Best	*Cascaded*	*PrStSu+SM*	82.9	71.7	76.9
Arabic	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	77.5	88.2	82.5
Mexicanero	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	77.9	81.0	79.4
Nahuatl	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	60.8	74.6	67.0
Wixarika	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	82.7	70.9	76.4
Mayo	*AG-LI-Auto*	*Standard*	*PrStSu+SM*	78.4	79.6	78.8
	AG-LI-Best	*Cascaded*	*PrStSu+SM*	82.9	78.8	80.8
English	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	75.3	78.5	76.9
German	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	81.3	76.0	78.6
Finnish	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	77.5	68.6	72.8
Estonian	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu+SM*	63.9	82.5	72.0
	AG-SS-Best	*Scholar-Seeded*	PrStSu+Co+SM	78.1	76.8	77.4
Turkish	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu2a+SM*	83.7	71.0	76.8
Zulu	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	85.0	65.3	73.8
Japanese	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu+SM*	82.3	77.6	79.9
	AG-SS-Best	*Scholar-Seeded*	*PrStSu*	79.3	80.7	80.0
Georgian	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu+SM*	84.3	67.9	75.2
	AG-SS-Best	*Scholar-Seeded*	*PrStSu*	79.6	72.5	75.9
Arabic	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	76.8	88.9	82.4
Mexicanero	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	82.9	82.1	82.5
Nahuatl	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	63.3	76.1	69.1
Wixarika	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	81.1	74.9	77.9
Mayo	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu2a+SM*	82.0	75.0	78.4
	AG-SS-Best	*Scholar-Seeded*	*PrStSu+SM*	84.4	78.7	81.5

Table 3.9: The performance of our automatically selected configuration versus the oracle performance (BPR). The upper part reports the language-independent performance (*AG-LI-Auto* and *AG-LI-Best*). The lower part reports the *Scholar-Seeded* performance (*AG-SS-Auto* and *AG-SS-Best*).

Language	Setup	Setting	Grammar	Prec.	Recall	F1
English	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	86.8	84.5	85.6
	AG-LI-Best	*Standard*	*PrStSu+SM*	87.7	85.0	86.3
German	*AG-LI-Auto*	*Standard*	*PrStSu+SM*	91.2	78.9	84.6
	AG-LI-Best	*Cascaded*	*PrStSu+SM*	90.9	79.2	84.7
Finnish	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	88.0	69.5	77.6
Estonian	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	79.7	90.8	84.9
	AG-LI-Best	*Standard*	*PrStSu+Co+SM*	88.9	86.1	87.5
Turkish	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	92.8	56.0	69.9
Zulu	*AG-LI-Auto*	*Cascaded*	*PrStSu+SM*	93.4	50.4	65.5
	AG-LI-Best	*Standard*	*PrStSu2b+SM*	93.0	55.9	69.8
Japanese	*AG-LI-Auto & AG-LI-Best*	*Cascaded*	*PrStSu+SM*	91.0	82.4	86.5
Georgian	*AG-LI-Auto*	*Standard*	*PrStSu+SM*	88.4	65.9	75.5
	AG-LI-Best	*Cascaded*	*PrStSu+SM*	88.8	67.8	76.9
Arabic	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	88.1	88.7	88.4
Mexicanero	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	91.2	89.0	90.1
Nahuatl	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	81.4	85.6	83.4
Wixarika	*AG-LI-Auto & AG-LI-Best*	*Standard*	*PrStSu+SM*	85.9	75.7	80.4
Mayo	*AG-LI-Auto*	*Standard*	*PrStSu+SM*	88.5	87.7	88.1
	AG-LI-Best	*Cascaded*	*PrStSu+SM*	89.5	87.5	88.5
English	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	88.5	85.1	86.7
German	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	90.2	80.4	85.0
Finnish	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	87.3	71.0	78.3
Estonian	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu+SM*	78.4	90.2	83.9
	AG-SS-Best	*Scholar-Seeded*	*PrStSu+Co+SM*	89.2	86.0	87.6
Turkish	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu2a+SM*	91.0	54.8	68.4
Zulu	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	88.9	66.4	76.0
Japanese	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	91.5	81.4	86.1
Georgian	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu+SM*	90.0	64.8	75.3
	AG-SS-Best	*Scholar-Seeded*	*PrStSu*	87.0	68.2	76.5
Arabic	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	87.5	89.4	88.4
Mexicanero	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	92.3	90.7	91.5
Nahuatl	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	81.2	87.5	84.2
Wixarika	*AG-SS-Auto & AG-SS-Best*	*Scholar-Seeded*	*PrStSu+SM*	84.7	79.5	82.0
Mayo	*AG-SS-Auto*	*Scholar-Seeded*	*PrStSu2a+SM*	91.1	86.0	88.5
	AG-SS-Best	*Scholar-Seeded*	*PrStSu+SM*	91.8	88.2	89.9

Table 3.10: The performance of our automatically selected configuration versus the oracle performance (EMMA-2). The upper part reports the language-independent performance (*AG-LI-Auto* and *AG-LI-Best*). The lower part reports the *Scholar-Seeded* performance (*AG-SS-Auto* and *AG-SS-Best*).

and *AG-SS-Auto* result in lower gaps of 1.6% and 2.1% in the average F1-scores, respectively, as compared to the corresponding oracle configurations.

3.5.4 Comparison to State-of-the-Art

We next compare the performance of our morphological-segmentation framework, MorphAGram, to two strong baselines, *Morfessor* and *MorphoChain* (Section 3.5.1). We report the results in Tables 3.11 and 3.12 using the BPR and EMMA-2 metrics, respectively, in terms of F1-score.

Language	Language-Independent Systems			*Scholar-Seeded* and Oracle Systems		
	Baselines		*MorphAGram (Auto)*		*MorphAGram (Oracle)*	
	Morfessor	*MorphoChain*	*AG-LI-Auto*	*AG-SS-Auto*	*AG-LI-Best*	*AG-SS-Best*
English	75.8	69.5	74.8	**76.9**	75.5	**76.9**
German	73.1	64.0	78.1	**78.6**	78.1	**78.6**
Finnish	62.9	55.7	71.6	**72.8**	71.6	**72.8**
Estonian	68.3	61.4	73.9	72.0	**77.4**	**77.4**
Turkish	64.9	60.6	**79.8**	76.8	**79.8**	76.8
Zulu	47.6	42.2	62.7	**73.8**	67.0	**73.8**
Japanese	79.6	61.8	79.8	79.9	79.8	**80.0**
Georgian	65.0	64.2	75.0	75.2	**76.9**	75.9
Arabic	78.2	77.1	**82.5**	82.4	**82.5**	82.4
Mexicanero	71.0	68.5	79.4	**82.5**	79.4	**82.5**
Nahuatl	60.3	56.1	67.0	**69.1**	67.0	**69.1**
Wixarika	72.9	38.7	76.4	**77.9**	76.4	**77.9**
Mayo	65.5	40.5	78.8	78.4	80.8	**81.5**
Average	68.1	58.5	75.4	76.6	76.3	**77.3**

Table 3.11: The performance of *MorphAGram* versus *Morfessor* and *MorphoChain* (BPR F1-score). The best overall result per language is in **bold**. The best language-independent result per language is underlined.

Considering the BPR metric, our scholar-seeded setup, *AG-SS-Auto*, outperforms our fully unsupervised setup, *AG-LI-Auto*, in nine languages and on average, achieving an average relative error reduction of 5.1%. As for the comparison to the baselines, *AG-LI-Auto* outperforms both *Morfessor* and *MorphoChain* when evaluated on all the experimental languages except English, with average relative error reductions of 22.8% and 40.7%, respectively. In the case of English,

Language	Language-Independent Systems			*Scholar-Seeded* and Oracle Systems		
	Baselines		*MorphAGram (Auto)*		*MorphAGram (Oracle)*	
	Morfessor	*MorphoChain*	*AG-LI-Auto*	*AG-SS-Auto*	*AG-LI-Best*	*AG-SS-Best*
English	86.0	82.5	85.6	**86.7**	86.3	**86.7**
German	81.0	73.9	84.6	**85.0**	84.7	**85.0**
Finnish	73.1	68.9	77.6	**78.3**	77.6	**78.3**
Estonian	83.7	75.1	84.9	83.9	87.5	**87.6**
Turkish	61.2	61.1	**69.9**	68.4	**69.9**	68.4
Zulu	52.3	55.9	65.5	**76.0**	69.8	**76.0**
Japanese	85.8	76.3	**86.5**	86.1	**86.5**	86.1
Georgian	69.4	69.3	75.1	74.9	**76.4**	76.0
Arabic	85.7	85.3	**88.4**	**88.4**	**88.4**	**88.4**
Mexicanero	86.7	86.8	90.1	**91.5**	90.1	**91.5**
Nahuatl	80.9	81.0	83.4	**84.2**	83.4	**84.2**
Wixarika	73.3	62.4	80.4	**82.0**	80.4	**82.0**
Mayo	80.7	78.1	88.1	88.5	88.5	**89.9**
Average	76.9	73.6	81.5	82.6	82.3	**83.1**

Table 3.12: The performance of *MorphAGram* versus *Morfessor* and *MorphoChain* (EMMA-2 F1-score). The best overall result per language is in **bold**. The best language-independent result per language is underlined.

Morfessor outperforms *AG-LI-Auto* by absolute 1.0%, while it comes second to *AG-SS-Auto* by absolute 1.1%.

The biggest and smallest gaps between *AG-LI-Auto* and *Morfessor* occur in the cases of Turkish and Japanese, respectively, where *AG-LI-Auto* achieves relative error reductions of 42.2% and 0.6%, respectively. On the other hand, the biggest and smallest gaps between *AG-LI-Auto* and *MorphoChain* occur in the cases of Mayo and English, respectively, where *AG-LI-Auto* achieves relative error reductions of 64.3% and 17.7%, respectively.

Considering the EMMA-2 metric, we observe similar patterns to those of the BPR metric, where *AG-LI-Auto* outperforms both *Morfessor* and *MorphoChain* when evaluated on all the experimental languages except English, with average relative error reductions of 20.1% and 30.1%, respectively. However, the biggest gaps between *AG-LI-Auto* and the baselines occur in the cases of polysynthetic languages, where *AG-LI-Auto* achieves a relative error reduction of 38.1% as compared to *Morfessor*

in the case of Mayo and a relative error reduction of 48.0% as compared to *MorphoChain* in the case of Wixarika.

It is worth noting that models that tend to consistently either under-segment or over-segment across the whole lexicon achieve significantly better EMMA-2 scores than the corresponding 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 F1-score from 40.5%, when using BPR, to 78.1%, when using EMMA-2, when evaluating *MorphoChain* on Mayo, where *MorphoChain* does under-segmentation with 100% precisions and low recalls at the detection of common affixes such as *ka*, *su* and *wa*.

One interesting finding is the ability of MorphAGram to handle polysynthetic languages, where a word may contain several morphemes, in low-resource setups of about 1,000 available words. Table 3.13 reports the performance of *MorphAGram* versus four supervised neural systems by Kann et al. (2018), namely *S2S* (seq2seq), *CRF*, *BestMTT* (the best multi-task training system) and *BestDA* (the best data-augmentation system), in terms of BPR F1-score when evaluating on *TEST*.

Language	Supervised Systems				*MorphAGram*			
	S2S	*CRF*	*BestMTT*	*BestDA*	*AG-LI-Auto*	*AG-SS-Auto*	*AG-LI-Best*	*AG-SS-Best*
Mexicanero	86.2	86.4	**87.9**	86.8	78.0	79.5	78.0	79.5
Nahuatl	72.7	**74.9**	73.9	73.2	72.3	74.4	73.6	74.4
Wixarika	79.6	79.3	80.2	**81.6**	76.8	78.6	76.8	78.6
Mayo	77.3	77.4	80.8	79.2	81.0	80.4	**81.1**	80.4

Table 3.13: The performance of *MorphAGram* versus the supervised neural systems by Kann et al. 2018) (BPR F1-score). The best result per language is in **bold.**

MorphAGram outperforms the supervised neural systems by Kann et al. (2018) when evaluated on Mayo using the same training and evaluation sets, with the main difference that we do not use the gold segmentation for training. In the case of Nahuatl, *AG-SS-Auto* is only 0.5% behind the best supervised system, *CRF*, while the gaps in the cases of Mexicanero and Wixarika are relatively small given the supervised nature of the baselines.

3.5.5 Impact of Linguistic Priors

Table 3.14 reports the morphological-segmentation performance with the incorporation of linguistic priors into the *PrStSu+SM* grammar in the form of a grammar definition, for Japanese, and linguist-provided affixes, for Georgian and Arabic. The results are compared to those of the corresponding regular settings using the BPR and EMMA-2 metrics. The use of linguistic priors consistently improves the performance in all the settings, and all the improvements are statistically significant for $p\text{-}value < 0.01$.

Language	Setting	BPR			EMMA-2		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Japanese	*Standard*	81.7	77.9	79.8	91.0	81.9	86.2
	Cascaded	81.5	78.2	79.8	91.0	82.4	86.5
	Scholar-Seeded	82.3	77.6	79.9	91.5	81.4	86.1
	Standard-LS	**83.1**	**79.0**	**81.0**	**91.7**	82.4	**86.8**
	Cascaded-LS	82.4	78.9	80.6	91.4	**82.5**	86.7
	Scholar-Seeded-LS	83.0	78.6	80.8	**91.7**	82.3	**86.8**
Georgian	*Standard*	82.0	69.1	75.0	88.4	65.9	75.5
	Cascaded	82.9	71.7	76.9	88.8	67.8	76.9
	Scholar-Seeded	84.3	67.9	75.2	**90.0**	64.8	75.3
	Scholar-Seeded-Ling	**84.6**	**82.3**	**83.5**	88.2	**78.5**	**83.1**
Arabic	*Standard*	77.5	88.2	82.5	88.1	88.7	88.4
	Cascaded	76.3	86.4	81.1	88.1	86.8	87.4
	Scholar-Seeded	76.8	88.9	82.4	87.5	89.4	88.4
	Scholar-Seeded-Ling	**81.4**	**96.2**	**88.2**	**89.0**	**96.5**	**92.6**

Table 3.14: The performance on Japanese, Georgian and Arabic with and without the use of linguistic priors within the *PrStSu+SM* grammar (BPR and EMMA-2). LS = Language-specific grammar. Ling = Linguist-provided affixes. The best result per language-metric pair is in **bold**.

In the case of Japanese, the use of a language-specific grammar (Figure 3.4) leads to the best performance in terms of precision, recall and F1-score, achieving relative error reductions of 6.0%, 4.2% and 4.5% in BPR F1-score in the *Standard*, *Cascaded* and *Scholar-Seeded* settings, respectively, with corresponding EMMA-2 relative error reductions of 4.2% 1.5% and 4.5%, respectively.

In the case of Georgian, the use of linguist-provided affixes yields the best results, with relative error reductions of 33.2% and 31.5% in BPR and EMMA-2 F1-score, respectively, as compared to the regular *Scholar-Seeded* setting that uses affixes of lower quality. However, the regular *Scholar-Seeded* setting achieves the best precision when evaluated using EMMA-2.

A similar pattern is seen in the case of Arabic, where the use of linguist-provided affixes yields the best performance in terms of precision, recall and F1-score, achieving relative error reductions of 32.9% and 35.9% in BPR and EMMA-2 F1-score, respectively, as compared to the *Scholar-Seeded* setting.

The use of linguist-provided affixes impacts recall more than precision, in both Georgian and Arabic, as the sampler gets informed about the most common affixes in the underlying language, which represent the majority of the affixes seen in the gold segmentation. However, precision also improves as the probability of utilizing existing production rules that represent the seeded affixes is usually higher than the probability of expanding new subtrees representing unseen affixes.

3.5.6 Performance of Multilingual Morphological Segmentation

We experiment with the following three multilingual setups:

- **Finnic-Uralic**: We combine the Finnish and Estonian lexicons as both are Uralic languages that belong to the Finnic language family. We test low-resource setups where we combine 500, 1,000, 5,000 and 10,000 words from each language.

  Mexicanero+Nahuatl: We combine the Mexicanero and Nahuatl lexicons as they are the closest two polysynthetic languages in our set of experimental languages, where Mexicanero is sometimes regarded as a dialect of Nahuatl.

  Uto-Aztecan: We combine the lexicons of the four Uto-Aztecan polysynthetic languages we experiment with (Mexicanero, Nahuatl, Wixarika and Mayo).

We examine our low-resource multilingual setups using the *Standard PrStSu+SM* configuration and report the results in Table 3.15 using the BPR and EMMA-2 metrics.

Language	Training Setup	BPR			EMMA2		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Finnish	Finnish (500)	59.3	71.6	**64.9**	70.8	76.9	**73.7**
Finnish	Finnic-Uralic (500)	56.2	70.4	62.5	69.2	75.3	72.1
Finnish	Finnish (1,000)	59.7	71.7	**65.1**	71.5	76.7	**74.0**
Finnish	Finnic-Uralic (1,000)	56.9	70.4	62.9	70.1	75.2	72.6
Finnish	Finnish (5,000)	62.7	71.6	**66.9**	74.8	75.3	**75.0**
Finnish	Finnic-Uralic (5,000)	61.8	69.2	65.3	75.3	72.6	73.9
Finnish	Finnish (10,000)	67.1	71.7	**69.3**	79.3	74.2	**76.6**
Finnish	Finnic-Uralic (10,000)	68.4	67.9	68.1	81.5	70.8	75.8
Estonian	Estonian (500)	32.6	88.2	47.5	41.5	93.9	57.6
Estonian	Finnic-Uralic (500)	34.1	86.4	**48.9**	45.8	92.6	**61.2**
Estonian	Estonian (1,000)	33.2	87.8	48.2	43.2	93.1	59.0
Estonian	Finnic-Uralic (1,000)	35.6	87.3	**50.6**	47.0	92.2	**62.3**
Estonian	Estonian (5,000)	44.2	87.2	58.4	56.8	92.5	70.2
Estonian	Finnic-Uralic (5,000)	45.7	86.1	**59.7**	59.4	91.5	**72.0**
Estonian	Estonian(10,000)	57.1	85.4	**68.4**	71.4	91.7	**80.3**
Estonian	Finnic-Uralic (10,000)	56.8	85.3	68.2	71.9	91.0	**80.3**
Mexicanero	Mexicanro	77.9	81.0	**79.4**	91.2	89.0	90.1
Mexicanero	Mexicanero+Nahuatl	78.9	79.8	79.3	92.7	88.4	**90.4**
Mexicanero	Uto-Aztecan	77.5	77.2	77.4	92.4	86.3	89.2
Nahuatl	Nahuatl	60.8	74.6	**67.0**	81.4	85.6	83.4
Nahuatl	Mexicanero+Nahuatl	60.1	74.5	66.5	81.5	85.5	**83.5**
Nahuatl	Uto-Aztecan	59.6	72.7	65.5	80.8	83.6	82.1
Wixarika	Wixarika	82.7	70.9	**76.4**	85.9	75.7	**80.4**
Wixarika	Uto-Aztecan	79.9	68.4	73.7	85.3	71.8	78.0
Mayo	Mayo	78.4	79.6	**78.8**	88.5	87.7	**88.1**
Mayo	Uto-Aztecan	75.6	76.9	76.2	85.7	85.8	85.7

Table 3.15: The performance of the low-resource multilingual setups. The best result per language-setup pair is in **bold**. The improvements due to the use of a multilingual setup that are statistically significant for $p\text{-}value < 0.01$ are circled.

The Finnic-Uralic multilingual training setup improves the performance for Estonian when training on small datasets of 500 and 1,000 words, achieving relative error reductions of 2.5% and 4.5% in BPR F1-score, respectively, and corresponding larger EMMA-2 reductions of 8.6% and 8.0%, respectively. However, the improvement becomes statistically insignificant for $p\text{-}value < 0.01$

when increasing the sizes of the merged lexicons to 5,000 words, while the monolingual setup surpasses the multilingual one with further increasing the sizes of the merged lexicons to 10,000 words.

For the polysynthetic multilingual setups, none of the EMMA-2 improvements for Mexicanero and Nahuatl in the Mexicanero+Nahuatl multilingual training setup is statistically significant for $p\text{-}value < 0.01$, and thus the combination of the Mexicanero and Nahuatl lexicons does not benefit either language. On the other hand, the Uto-Aztecan setup consistently results in performance drops for the four polysynthetic languages.

In the cases where multilingual training helps, the performance improves due to an increase in precision. This suggests that the addition of data points from another related language might decrease the number of incorrectly expanded subtrees. Another observation is that combining the lexicons of too many languages, like in the case of the Uto-Aztecan setup, does not benefit any of the underlying languages as too many data points become misleading to the sampler and result in over-segmentation.

3.5.7 Learning Curves

We examine the performance of the *Standard* (*STD*) and *Scholar-Seeded* (*SS*) *PrStSu+SM* configurations on all the experimental languages except the low-resource polysynthetic ones when training on different sizes of 500, 1,000, 5,000, 10,000, 20,000, 30,000, 40,000 and 50,000 words. The learning curves are depicted in Figure 3.7 based on BPR F1-score.

At small training sets of 500 and 1,000 words, *SS* outperforms *STD* except in Arabic. With the addition of more training data, there are four possible scenarios: 1) *SS* consistently takes the lead (English, Finnish and Zulu); 2) *STD* and *SS* exchange positions and end up performing similarly at the largest experimental training sets (German and Georgian); 3) *STD* and *SS* exchange positions until one setting supersedes the other (*SS* in Turkish and *STD* in Estonian); and 4) *SS* and *STD* behave almost similarly across the different sizes (Japanese and Arabic).

In the cases of English and Arabic, the performance of *STD* consistently increases with the

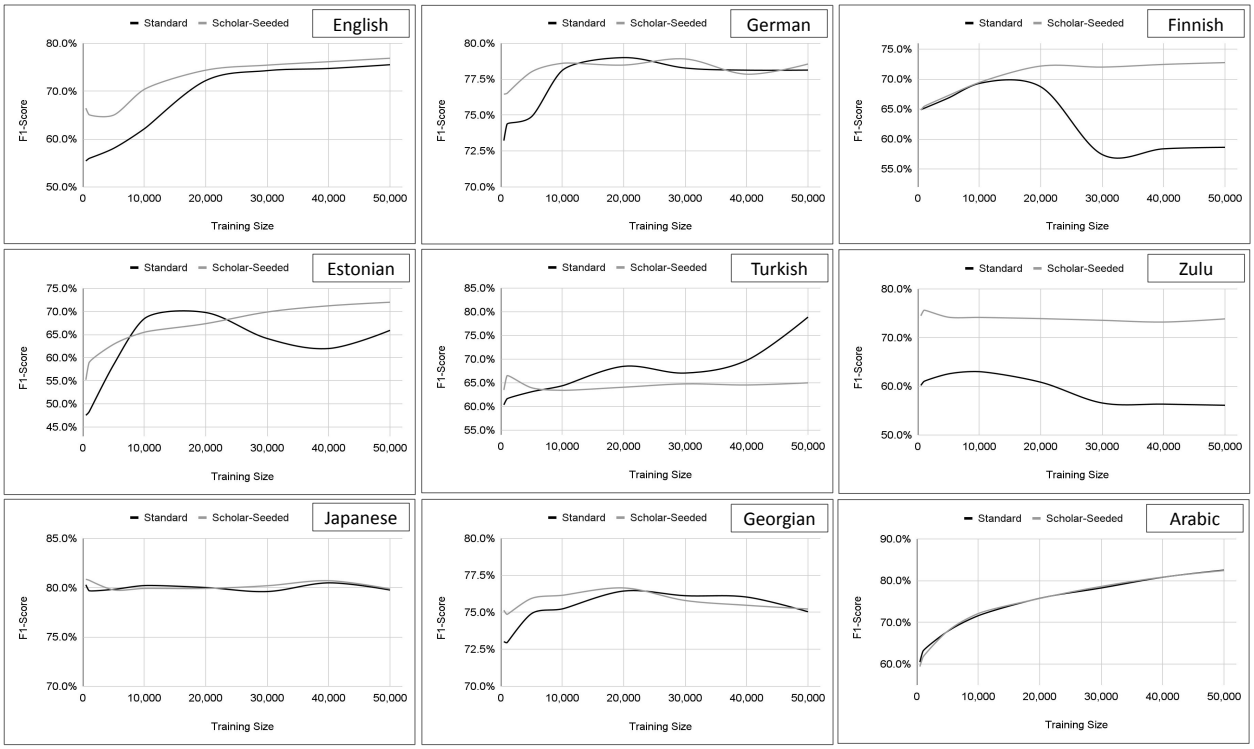


Figure 3.7: The learning curves of the *Standard* and *Scholar-Seeded PrStSu+SM* configurations (BPR F1-score)

addition of more training examples. Otherwise, the performance fluctuates with either an upward pattern (German, Estonian, Turkish and Georgian) or a downward one (Finnish and Zulu), while the pattern is nearly flat in the case of Japanese.

The possible downward pattern happens when a language tends to have a large number of morphs per word, and thus the addition of data points confuses the sampler, causing excessive unnecessary expansions of new subtrees. This explains the downward trends in Finish and Zulu, which are the languages with the highest average number of morphs per word (Figure 3.5a). On the other hand, the flat pattern in Japanese indicates early saturation, where the sampler expands the majority of the new subtrees after examining a few hundreds of words.

It is noteworthy to mention that the performance of *MorphAGram* when only using 500 words for training outperforms the performance of the baselines, *Morfessor* and *MorphoChain*, when they utilize 50,000 words for training in the cases of English, German, Zulu, Japanese and Georgian. This

is because MorphAGram learns well from a small amount of data, which is why it learns efficient morphological-segmentation models for the polysynthetic languages in low-resource setups.

3.5.8 Error Analysis of Morphological Segmentation

3.5.8.1 Error Analysis in the Fully Unsupervised Setting

Table 3.16 lists some examples of correctly and incorrectly segmented words by our main *MorphAGram* setups, *AG-LI-Auto* and *AG-SS-Auto*, for each experimental language. We next discuss the most noticeable phenomena based on the segmentation outputs of *MorphAGram*, *Morfessor* and *MorphoChain* and using the BPR metric.

English The performance of *MorphAGram*, *Morfessor* and *MorphoChain* noticeably differs across the six most common affixes, namely *'s*, *'*, *s* (both the nominal plural suffix and the verbal present suffix), *er*, *ed* and *ing*. The three systems achieve 100% F1-scores at the detection of the *'s* suffix, while they all tend to merge the *'* suffix with the preceding *s* suffix. However, both *Morfessor* and *MorphoChain* outperform *AG-LI-Auto* and *AG-SS-Auto* at the detection of the *s* suffix with F1-scores of 70.7% and 89.8% as opposed to 61.5% and 67.7%, respectively In contrast, *AG-LI-Auto* and *AG-SS-Auto* considerably outperform *Morfessor* and *MorphoChain* at the detection of the *er*, *ed* and *ing* suffixes with average F1-scores of 91.9% and 92.3%, respectively, as opposed to 67.7% by *Morfessor* and 80.7% by *MorphoChain*. Moreover, *MorphAGram* is able to detect several suffixes that both *Morfessor* and *MorphoChain* consistently fail to detect, such as *iz*, *or* and *at*. *MorphoChain* further fails to detect several suffixes, such as *ion*, *ant* and *ance*. On another hand, one common mistake by *MorphAGram* and *MorphoChain* is the segmentation of the ending *e* as a separate suffix. This is because in several cases, the removal of an ending *e* forms another valid word, e.g., as in *made* and *huge*.

German *AG-LI-Auto* and *AG-SS-Auto* are efficient at detecting the 10 most frequent affixes, which are all suffixes, with average F1-scores of 69.7% and 77.0%, respectively, as opposed to

Language	word	Gold Seg.	*AG-LI-Auto* Seg.	*AG-SS-Auto* Seg.
English	*dismemberment's* *foot-faulting* *peelers'* *necklace*	*dis+member+ment+'s* *foot+-+fault+ing* *peel+er+s+'* *neck+lace*	*dis+member+ment+'s* *foot+-+fault+ing* *peel+er+s'* *necklac+e*	*dis+member+ment+'s* *foot+-+fault+ing* *peel+er+s'* *necklace*
German	*abzugrenzen* *anfeuchtet* *aufenthalte* *verträglichkeiten*	*ab+zu+grenz+en* *an+feucht+et* *auf+ent+halt+e* *ver+träg+lich+keit+en*	*ab+zu+grenz+en* *an+feucht+et* *auf+enthalt+e* *verträg+lichkeit+en*	*ab+zu+grenz+en* *an+feucht+et* *auf+enthalt+e* *verträg+lich+keit+en*
Finnish	*taudinkuvat* *peruskorossa* *pressujen* *oikeudenhaku*	*taudi+n+kuva+t* *perus+koro+ssa* *pressu+j+en* *oike+ude+n+haku*	*taudi+n+kuva+t* *perus+koro+ssa* *pressu+jen* *oikeude+n+haku*	*taudi+n+kuva+t* *perus+koro+ssa* *press+ujen* *oikeude+n+haku*
Estonian	*täieõiguslikud* *päikesepaistelises* *ragisesid* *raudteejaamades*	*täie+õigus+liku+d* *päikese+paiste+lise+s* *ragise+sid* *raud+tee+jaama+des*	*täie+õigus+liku+d* *päikese+paiste+lise+s* *ragise+sid* *raudtee+jaama+des*	*täie+õigus+liku+d* *päikese+paiste+lise+s* *ragise+s+id* *raudtee+jaama+de+s*
Turkish	maksadımızı püniversitelerle *bozmasıydı* *türkerden*	maksad+ımız+ı püniversite+ler+le *boz+ma+sı+ydı* *türker+den*	maksad+ımız+ı püniversite+ler+le *boz+ması+ydı* *türker+den*	maksad+ımız+ı püniversite+ler+le *boz+ması+ydı* *türk+er+den*
Zulu	*naloya* *angafike* *ngathola* *bayibona*	*na+lo+ya* *a+nga+fik+e* *ng+a+thol+a* *ba+yi+bon+a*	*na+lo+ya* *a+nga+fike* *nga+thola* *ba+yibona*	*na+lo+ya* *a+nga+fik+e* *nga+thol+a* *ba+yi+bon+a*
Japanese	終わらせる 来る 散れません 惚れました	終わ+らせ+る 来+る 散+れ+ま+せん 惚れ+ま+した	終わ+らせ+る 来+る 散+れ+ません 惚れ+ました	終わ+らせ+る 来+る 散+れ+ません 惚れ+ました
Georgian	ნაჭამი სტადიონზე ვიზამ თვეშიც	ნა+ჭამ+ი სტადიონ+ზე ვ+ი+ზ+ამ თვე+ში+ც	ნა+ჭამ+ი სტადიონ+ზე ვიზ+ამ თვე+ში+ც	ნა+ჭამ+ი სტადიონ+ზე ვიზ+ამ თვე+შიც
Arabic	ومساعدته التحديات محفوظة وستقوم	و+مساعد+ت+ه ال+تحدي+ات محفوظ+ة و+س+ت+قوم	و+مساعد+ت+ه ال+تحدي+ات ة+حفوظ+م و+ست+قوم	و+مساعد+ت+ه ال+تحدي+ات ة+حفوظ+م و+ست+قوم
Mexicanero	*tikimpiyal* *nibolsaiyo* *titakwatikaá* *ukitasa*	*ti+kim+piya+l* *ni+bolsa+iyo* *ti+ta+kwa+ti+ka+á* *u+ki+tasa*	*ti+kim+piya+l* *ni+bolsa+iyo* *ti+ta+kwa+tikaá* *u+ki+ta+sa*	*ti+kim+piya+l* *ni+bolsa+iyo* *ti+takwa+tika+á* *u+ki+ta+s+a*
Nahuatl	*tikintlatlanilia* *onisiaw* *nankochtikate* *otinechtiak*	*ti+kin+tla+tlanilia* *o+ni+siaw* *nan+koch+tika+te* *o+ti+nech+tia+k*	*ti+kin+tla+tlanilia* *o+ni+siaw* *nan+kochtika+te* *o+ti+ne+chtia+k*	*ti+kin+tla+tlanilia* *o+ni+siaw* *nan+kochtika+te* *oti+nech+tia+k*
Wixarika	*kene'a'eriwatü* *püneniwe* *tsitepa'u* *neputatsukaxi*	*ke+ne+'a+'eriwa+tü* *pü+ne+niwe* *tsi+te+p+a+'u* *ne+pu+ta+tsuka+xi*	*ke+ne+'a+'eriwa+tü* *pü+ne+niwe* *tsi+tepa+'u* *ne+pu+tatsu+ka+xi*	*ke+ne+'a+'eriwa+tü* *pü+ne+niwe* *tsi+te+pa'u* *ne+puta+tsu+ka+xi*
Mayo	*usimpo* *techowatuari* *ßohoßareka* *sikaye'wi*	*usi+m+po* *techowa+tua+ri* *ßoho+ßa+re+ka* *sika+ye+'wi*	*usi+m+po* *techowa+tua+ri* *ßoho+ßare+ka* *sika+ye+'wi*	*usi+m+po* *techowa+tua+ri* *ßoho+ßare+ka* *si+ka+ye+'wi*

Table 3.16: Samples of correct and incorrect morphological-segmentation examples

considerably lower F1-scores of 53.8% and 53.4% by *Morfessor* and *MorphoChain*, respectively. Regarding the prefixes, both *MorphAGram* and *Morfessor* can recognize the prefixes *an*, *ab* and *auf* with high F1-scores of at least 83.3% despite their relatively low frequencies, while *MorphoChain* is significantly less efficient at detecting the three affixes, with a maximum F1-score of 71.4%. On another hand, *MorphAGram* is able to recognize several affixes that the other systems always fail to detect. For instance, *MorphAGram* is able to detect the affixes *end*, *recht* and *et* with F1-scores of at least 50.0%, while *Morfessor* consistently fails to detect *end* and *et*, and *MorphoChain* does not recognize *end* and *recht*. However, the three systems tend to generally under-segment in the case of German.

Finnish *AG-LI-Auto* and *AG-SS-Auto* significantly outperform *Morfessor* and *MorphoChain* at the detection of the four most common morphs, which are all one-letter ones, namely *i*, *n*, *t* and *a*, with average F1-scores of 57.9% and 58.9%, respectively, as opposed to 34.8% by *Morfessor* and 40.6% by *MorphoChain*. These morphs constitute 17.1% of all the morphs, which is the main reason behind the superiority of *MorphAGram* in the case of Finnish. In addition, *MorphAGram* is efficient at detecting several long morphs of three or more letters that *Morfessor* and *MorphoChain* usually fail to detect, such as *ssa*, *ksi*, *ssä* and *stä*. In addition, the three systems suffer at the detection of some frequent morphs, such as *j*, with a maximum F1-score of 3.8% by *AG-SS-Auto* and *Morfessor*, and *u*, with a maximum F1-score of 16.3% by *AG-SS-Auto*. However, similarly to German, the three systems tend to generally under-segment in the case of Finnish.

Estonian *AG-LI-Auto* outperforms *AG-SS-Auto* at the detection of the 10 most frequent morphs, achieving an average F1-score of 87.7%, as opposed to 77.2% by *AG-SS-Auto*, while *Morfessor* and *MorphoChain* perform relatively similar to *AG-SS-Auto*. However, the performance of the three systems on the most common morphs in Estonian is significantly better than on those of the other languages. One reason is that only two morphs in the 10 most frequent ones are relatively ambiguous, namely *ks* and *ga*, where their surface forms represent morphs 52.2% and 45.9% of the time they appear, respectively. This is in addition to the fact that Estonian tends to have a small

number of morphs per word, compared to the other experimental languages (Section 3.4). On another hand, *AG-SS-Auto* consistently fails to detect the suffixes *sid* and *des*, where it tends to segment them as *s+id* and *de+s*, respectively, while *AG-LI-Auto* achieves F1-scores of 98.2% and 90.3% at the detection of the two suffixes, respectively.

Turkish *MorphAGram* shows better detection of one-letter affixes than *Morfessor* and *MorphoChain*. For instance, *AG-LI-Auto* and *AG-SS-Auto* detect the most common three one-letter affixes, namely *i*, *ı* and *t*, with average F1-scores of 57.1% and 63.1%, respectively, as opposed to 17.5% and 32.4% by *Morfessor* and *MorphoChain*, respectively. However, the detection of these affixes remains a challenge because they are highly ambiguous as their surface forms usually appear as part of longer morphs. On another hand, the three systems detect the most frequent morph, *ler* (a plural suffix), with 100% precisions. However, *ler* is part of longer morphs 25.1% of the time it appears, which lowers the recalls to 58.8%, 32.0% and 64.5% by *MorphAGram*, *Morfessor* and *MorphoChain*, respectively. Most of such under-segmentation errors occur when *ler* is followed by a vowel. Another interesting case is the morph *ma*. Despite the fact that *ma* is the sixth most frequent morph in the data, the three systems tend to merge it with *sı* due to the frequent occurrence of *ması*. Another case is the suffix *den*, where *AG-LI-Auto* identifies it correctly with a 100.0% F1-score despite its high degree of ambiguity, where *den* is a suffix only 47.5% of the time it occurs, while *Morfessor* and *MorphoChain* achieve noticeably lower corresponding F1-scores of 62.5% and 88.5%, respectively. However, the three systems tend to generally under-segment in the case of Turkish.

Zulu One observed phenomenon in the case of Zulu is that our *MorphAGram* setups vary widely in their performance. For instance, while *AG-SS-Auto* is able to detect the two most frequent affixes, namely *a* and *e*, with F1-scores of 75.0% and 73.2%, respectively, *AG-LI-Auto* achieves significantly lower F1-scores of 22.5% and 37.9%, respectively. However, *AG-SS-Auto* is more efficient at detecting the two affixes when they appear at the end of a word. In contrast, *Morfessor* and *MorphoChain* achieve lower F1-scores on the two morphs, which in turn affects the performance

of both systems on Zulu as the two affixes constitute 23.1% of the morphs. Another interesting case is the prefix *nga*, which *AG-LI-Auto* and *AG-SS-Auto* are able to identify with F1-scores of 79.0% and 72.7%, respectively, which is supported by its low degree of ambiguity and high frequency. However, both *Morfessor* and *MorphoChain* fail to detect the prefix most of the time, with low F1-scores of 21.8% and 28.1%, respectively. On another hand, *MorphAGram* consistently fails to detect the affix *ng* as its surface form is usually part of other affixes such as *nga* and *ngi*, while *Morfessor* and *MorphoChain* can detect the affix with recalls of 7.0% and 35.2%, respectively. Another observation is that *AG-SS-Auto* achieves a 100% F1-score on several affixes that *AG-LI-Auto* and the baselines always fail to detect, such as *el*, *is*, *bon* and *ek*.

Japanese *MorphAGram*, *Morfessor* and *MorphoChain* achieve relatively low F1-scores up to 59.6% at the detection of the two most frequent morphs, namely な and い. Moreover, *MorphAGram* always fails at the detection of ま, the third most frequent morph, on which both *Morfessor* and *MorphoChain* achieve a relatively low F1-score of 22.2%. Moreover, *MorphoChain* always fails to detect several morphs, such as った and させ. On another hand, there are several affixes that only *AG-LI-Auto* can detect with 100.0% F1-scores, such as る and 来, which rank fourth and sixteenth in terms of frequency, respectively. However, the three systems consistently fail at the detection of the suffixes せん and した. It is also observed that most of the errors made by *Morfessor* and *MorphoChain* are due to under-segmentation; especially, 15 out of the 20 most frequent morphs are one-letter ones. However, the under-segmentation in *Morfessor* is less excessive and allows for the detection of long morphs of three or more letters, such as かった and しょう.

Georgian *AG-LI-Auto* and *AG-SS-Auto* significantly outperform *Morfessor* and *MorphoChain* at the detection of the most common one-letter morphs, namely ი, ა, ხ, ე, მ, ო and ვ, with average F1-scores of 57.4% and 57.9%, respectively, as opposed to 37.3% by *Morfessor* and 42.0% by *MorphoChain*. However, these morphs are highly ambiguous and difficult to detect. For instance, the three systems achieve low recalls, up to 50.6%, at detecting the two most frequent morphs, namely ი and ა, where their surface forms appear as part of longer morphs, such as ში, და and

გა. *MorphAGram* can also detect several affixes that *Morfessor* and *MorphoChain* consistently fail to detect, such as ავ, ელ and ილ. On another hand, the suffixes ზე and ად are the mostly recognized affixes among the most frequent ones, with up to a 100.0% F1-score by *MorphAGram* at the detection of ზე. However, the three systems tend to generally under-segment in the case of Georgian.

For the performance of *AG-LI-Auto* with respect to the POS categories assigned to the words in the gold standard, *AG-LI-Auto* achieves F1-scores of 73.4%, 68.5%, 82.8% and 83.6% on the nominal, verbal, numeral and "other" categories, respectively. However, while the detection of the verbal category witnesses the highest precision of 95.6%, it suffers the lowest recall of 53.4%. In contrast, the detection of the nominal category has a high recall of 77.3%.

Arabic *MorphAGram* outperforms *Morfessor* and *MorphoChain* at the detection of the common affixes ة and ت, which rank third and fourth in terms of frequency, respectively, and constitute 9.9% of the morphs. *AG-LI-Auto* and *AG-SS-Auto* achieve average F1-scores of 89.1% and 89.4% on the two morphs, respectively, as opposed to 59.4% and 62.3% by *Morfessor* and *MorphoChain*, respectively. However, the three systems behave similarly at the detection of the two most common affixes, namely ال and و. On another hand, one common mistake by *MorphAGram* and *MorphoChain* is the segmentation of the beginning م. This is because many Arabic adjectives start with م, where it is actually part of the stem. In addition, *AG-LI-Auto* and *MorphoChain* tend to over-segment ك when it is part of a stem, confusing it with the prefix ك. Moreover, *AG-SS-Auto* consistently fails to detect the verbal prefix س, where it merges it with the following morph. It is also observed that some segmentation errors are actually correct regardless of the context, while the gold segmentation is based on the contexts seen in the PATB. An example is the word تشبه, which means either ت+شبه (she/it looks like) or تشبه (resembling).

Mexicanero *MorphAGram* outperforms *Morfessor* and *MorphoChain* at the detection of the eight most common morphs, where *AG-LI-Auto* and *AG-SS-Auto* achieve average F1-scores of 85.1% and 84.5%, respectively, as opposed to 26.5% by *Morfessor* and 9.1% by *MorphoChain*. However,

none of the systems is able to detect the ninth most common morph, *ka*. Also, several morphs are only recognizable by *MorphAGram*, such as *ki* and *nich*, which rank first and sixth in terms of frequency, respectively. The errors made by *Morfessor* and *MorphoChain* are mainly due to under-segmentation, where the surface forms of several morphs are part of longer ones, which in turn results in 100% precisions and low recalls at the detection of those morphs.

Nahuatl *AG-LI-Auto* and *AG-SS-Auto* significantly outperform *Morfessor* and *MorphoChain* at the detection of the 10 most common morphs, with average F1-scores of 70.0% and 77.4% as opposed to 24.2% and 5.5%, respectively. Moreover, *MorphAGram* is the only system that can detect the morphs *k* and *tla*, which rank second and sixth in terms of frequency, respectively. However, none of the systems can achieve an F1-score of 100.0% at the detection of any of the seven most common morphs. On the other hand, *MorphoChain* consistently fails to detect 15 morphs out of the 20 most common ones due to severe under-segmentation. However, the three systems tend to generally under-segment in the case of Nahuatl.

Wixarika *AG-LI-Auto* and *AG-SS-Auto* detect the two most frequent affixes, namely *pü* and *ne*, efficiently with average F1-scores of 70.7% and 97.5%, respectively. In contrast, *Morfessor* and *MorphoChain* detect the two affixes with lower average F1-scores of 63.3% and 15.3%, respectively. Moreover, *MorphoChain* consistently fails to detect the next two most frequent affixes, namely *ti* and *ka*. In addition, while *MorphAGram* can detect four out of the 10 most frequent morphs with F1-scores of at least 80.0%, *MorphoChain* shows a high degree of under-segmentation, where it achieves a 0.0% F1-score at the detection of seven out of the 10 most frequent morphs and up to 18.2% in F1-score on the rest of the morphs. However, none of the systems can detect the common affixes *p*, *e* and *r*, which rank fifth, sixth and eleventh in terms of frequency, respectively. However, the three systems tend to generally under-segment in the case of Wixarika.

Mayo *MorphAGram* outperforms *Morfessor* and *MorphoChain* at the detection of the two most common morphs, namely *m* and *k*, where *AG-LI-Auto* and *AG-SS-Auto* achieve average F1-scores

of 85.8% and 85.4%, respectively, as opposed to 53.5% and 0.0% by *Morfessor* and *MorphoChain*, respectively. In fact, *MorphoChain* achieves a 0.0% F1-score at the detection of seven out of the 10 most frequent morphs, which constitute 24.1% of the morphs. Moreover, *MorphAGram* achieves a 100% F1-score on several morphs of high degrees of ambiguity, such as *po* and *ri*, whose surface forms represent the morphs only 57.1% and 43.8% of the time they occur, respectively. However, none of the systems is able to detect the morphs *ßa* and *re*.

3.5.8.2 Error Analysis when Using Linguistic Priors

We next analyze the main lines of improvements due to the incorporation of linguistic priors for Japanese, Georgian and Arabic.

Japanese The use of linguistic priors in the form of a grammar definition improves the detection of several affixes. For example, the affixes れば, 踊, 通, 定め, 終わ, り and られ receive relative error reductions of more than 50.0% in F1-score. Two interesting cases are the affixes れば and した, which the regular *Standard* setting consistently fails to detect, but they can be detected through the use of the Japanese language-specific grammar. In addition, the language-specific grammar outperforms the language-independent one at the detection of stems because of the explicit modeling of compounding. For instance, the detection of the common stem られ (*be*) improves by a relative error reduction of 53.8% in F1-score.

Georgian The incorporation of linguist-provided affixes improves the recognition of the 10 most common affixes from 52.2% to 81.1% in F1-score on average. Moreover, the detection of the affixes ზე, ში, ებ, მო, გა, დ, ობ, ის and ს significantly improves by relative error reductions of more than 80.0% in F1-score. These affixes constitute 19.8% of the morphs in Georgian, which explains the noticeable improvements with the incorporation of linguistic priors. However, the performance on the affixes უ and თ, for instance, drops by absolute F1-scores of 30.7% and 2.5%, respectively, although they are provided as linguistic priors. This is because the surface forms of these affixes are part of other longer ones that the sampler mistakenly picks.

Seeding linguist-provided affixes achieves F1-scores of 81.6%, 80.4%, 86.3% and 88.3% on the nominal, verbal, numeral and "other" categories, respectively, which is an average relative error reduction of 29.3% as compared to the *Standard* setting. In fact, the verbal and nominal categories benefit from the linguistic priors the most with relative error reductions of 37.7% and 31.1%, respectively.

Arabic The use of linguistic priors in the form of linguist-provided affixes improves the detection of the 10 most common affixes from 86.9% to 95.2% in F1-score on average. In fact, the detection of several affixes improves significantly, where the affixes ال, و, ة, ب, ا, ين, ون and وا receive relative error reductions of at least 80.0% in F1-score. These affixes constitute 34.9% of the morphs, which explains the considerable improvements due to the use of linguistic priors. Two interesting cases are the suffix وا and the prefix س, which the regular *Standard* setting consistently fails to detect, while the use of linguist-provided affixes allows for their detection with F1-scores of 100.0% and 50.0%, respectively.

3.6 Conclusion

In this chapter, we introduced *MorphAGram*, a publicly available framework for unsupervised and minimally supervised morphological segmentation that is based on Adaptor Grammars (AGs), where PCFGs are utilized to model word structure.

We proposed several language-independent grammar definitions and defined three learning settings: *Standard*, *Scholar-Seeded* and *Cascaded*. While the *Standard* setting is fully unsupervised, the *Scholar-Seeded* one utilizes linguistic knowledge by seeding affixes that are generated from language resources into the grammars. The *Cascaded* setting is a fully unsupervised self-training approach that automatically learns linguistic knowledge before seeding it into the grammars.

We next proposed a new approach for the selection of the optimal configuration (a learning setting and a grammar) for an unseen language. This includes two setups: 1) *AG-LI-Auto*: a fully unsupervised setup; and 2) *AG-SS-Auto*: a minimally-supervised setup that applies the *Scholar-*

Seeded setting. In addition, we introduced new approaches for the incorporation of linguistic priors within AGs: 1) designing a language-specific grammar; and 2) seeding linguist-provided affixes of high quality. Moreover, we examined multilingual morphological segmentation in low-resource setups, where the lexicons of different related languages are combined.

We evaluated *MorphAGram* on 13 languages of diverse typologies, namely English, German, Finnish, Estonian, Turkish, Zulu, Japanese, Georgian, Arabic, Mexicanero, Nahuatl (Mexicano), Wixarika (Huichol) and Mayo (Yorem Nokki), where we utilized high-resource and low-resource setups. Our evaluation showed that *AG-LI-Auto* outperforms both *Morfessor* and *MorphoChain*, two strong baselines, with average relative error reductions of 22.8% and 40.7% in BPR F1-score, respectively. We also showed that linguistic priors help, where we achieved improvements upon the design of a language-specific grammar for Japanese and the seeding of linguist-provided affixes for Georgian and Arabic. Finally, in multilingual morphological segmentation, we showed performance gains for Estonian upon combining small Finnish and Estonian lexicons.

We conducted extensive analyses where we analyzed the morphological characteristics of the experimental languages and how the performance changes across datasets of various sizes. We also reported the most noticeable phenomena upon analyzing the segmentation output of *MorphAGram* as compared to the segmentation outputs by *Morfessor* and *MorphoChain* for each language. In addition, we analyzed the segmentation outputs upon incorporating linguistic priors for Japanese, Georgian and Arabic.

Chapter 4

Unsupervised Cross-Lingual Part-of-Speech Tagging

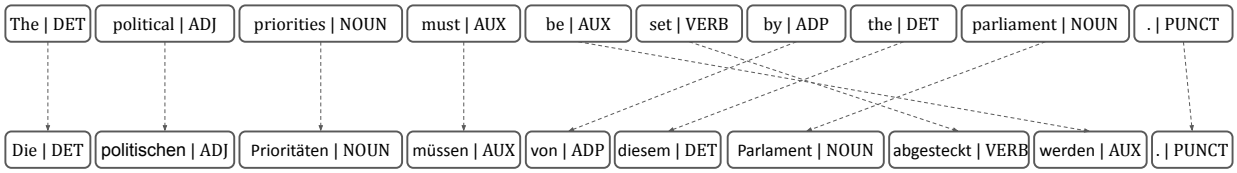
4.1 Overview

Unsupervised cross-lingual part-of-speech (POS) tagging via annotation projection relies on the use of parallel data to project POS tags from a source language for which a POS tagger is accessible onto a target language across word-level alignments. The projected tags then form the basis for learning a POS model of the target language.

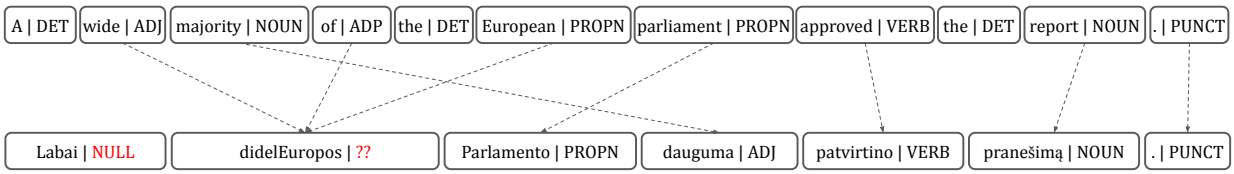
POS tagging via annotation projection has a long research history that investigates several unsupervised and semi-supervised approaches (Yarowsky et al., 2001; Fossum and Abney, 2005; Das and Petrov, 2011; Duong et al., 2013; Agić et al., 2015; Agić et al., 2016; Buys and Botha, 2016). All these approaches either use domain-appropriate and/or large parallel data, rely on multiple source languages to project from or exploit linguistic priors or annotations in the target language. On the contrary, we focus on fully unsupervised learning in truly low-resource scenarios, where we do not require access to domain-specific or large parallel data, parallel data of more than two languages nor linguistic information on the target side. We use the Bible as the translation source as it is available in a large number of languages, many of which are low-resource and meets our low-resource assumptions: small in size and out-of-domain with respect to the evaluation sets.

One major concern is that annotation projection suffers from several issues such as bad translation, alignment mistakes and translation phenomena that do not fit the assumptions of word-alignment models. Moreover, there is no one-to-one correspondence between the POS tags across two different languages as languages differ in their morphological and syntactic typologies. This could result in null alignments and noisy and unreliable annotations. As a result, a key consideration for all the previous approaches in the literature is how to obtain high-quality alignments and

projected annotations towards clean training instances in the target language.



(a) One-to-one English-to-German alignments



(b) Non-one-to-one English-to-Lithuanian alignments

Figure 4.1: Word-based alignment examples

Figure 4.1a shows an example of perfect one-to-one alignments from English to German, where each English word maps to one corresponding German word and vice versa. However, in practice, this is rarely the case. Null, one-to-many and many-to-one alignments are unsurprising. Figure 4.1b shows an example of aligning an English sentence to its Lithuanian translation, where some English words remain unaligned (*a* and *the*), while multiple English words map to one Lithuanian word (*wide*, *of* and *European* map to *didelEuropos*).

Our contribution is threefold:

- We present a framework for unsupervised cross-lingual POS tagging [1] in which we standardize the process of annotation projection in a robust approach that exploits and expands the best practices in the literature, where we aim to produce reliable annotations towards efficient POS models. This includes, but is not limited to, the use of bidirectional alignments, coupling token and type constraints on the target side and scoring the annotated sentences for the selection of reliable training instances (Section 4.2.1).

[1] https://github.com/rnd2110/unsupervised-cross-lingual-POS-tagging

- We use transformer-based contextualized word embeddings to train POS models based on the projected annotations, which is, to our knowledge, the first work that utilizes transformer-based contextualized embeddings for unsupervised cross-lingual POS tagging via annotation projection. We design a rich BiLSTM (Hochreiter and Schmidhuber, 1997) neural architecture that combines word embeddings, affix embeddings and word-cluster embeddings, along with special handling for the null assignments resulting from missing and rejected alignments and non-overlapping token and type constraints (Section 4.2.2).

- We conduct extensive evaluation and analysis using six commonly spoken source languages, namely English, Spanish, French, German, Russian and Arabic, and 14 typologically diverse target languages, namely, Afrikaans, Amharic, Basque, Bulgarian, Finnish, Georgian, Hindi, Indonesian, Kazakh, Lithuanian, Persian, Portuguese, Telugu and Turkish, for a total of 84 target-source language pairs, where we introduce a new gold-standard POS-labeled dataset for Georgian (Sections 4.3 and 4.4).

We show that our approach is highly efficient for POS tagging in a fully unsupervised setup, where it achieves an average POS accuracy of 75.5% across all the language pairs. We also demonstrate significant improvements over previous work, both unsupervised and semi-supervised. Additionally, we study ablation setups that lack some of the exploited computational and linguistic resources and show that our approach can still perform relatively well in such restricted settings (Section 4.4).

We show that our models are able to predict at least as many as 85.0% of the correct decisions made by the corresponding supervised ones in eight target languages, and thus annotation projection might provide an alternative to supervised learning, which is usually costly and time consuming. We also investigate how much manually labeled data is needed in order to develop supervised models comparable to our best unsupervised ones (Section 4.5).

Finally, we demonstrate that our approach outperforms zero-shot model transfer when the source and target languages are typologically dissimilar as zero-shot model transfer cannot generalize well across unrelated languages (Section 4.6).

This chapter contains our work on unsupervised cross-lingual POS tagging for truly low-resource scenarios (Eskander et al., 2020b) and extends it by providing deeper evaluation and analysis, along with the addition of two morphologically complex target languages, namely Georgian and Kazakh.

4.2 Methods

Our approach for unsupervised cross-lingual POS tagging is divided into two main phases:

1. generating POS annotations by performing cross-lingual projection via word-level alignments between a high-resource language for which a POS tagger is available and the underlying target language (Section 4.2.1); and

2. training a neural POS tagger for the target language based on the projected annotations (Section 4.2.2).

The overall pipeline is illustrated in Figure 4.2.

4.2.1 Cross-lingual Projection via Word Alignments

We describe below the steps of projecting the POS tags from a source language onto the target one via word-level alignments. An example of alignment and projection is shown in Figure 4.3, where English and Persian are the source and target languages, respectively. The example corresponds to verse *EXO 16:30*, *"So the people rested on the seventh day."*, where the word-alignment models (English-to-Persian and Persian-to-English) are trained on the entire Bible after the preliminary step of white-space tokenization.

White-Space Tokenization Starting with a sentence-aligned parallel text, we first perform white-space tokenization on both the source and target sides, where we separate punctuation marks and symbols into standalone tokens. We use *Stanza* [2] (Qi et al., 2020) to tokenize five of our six experimental source languages, namely English, Spanish, French, German and Russian, while we

[2] https://github.com/stanfordnlp/Stanza

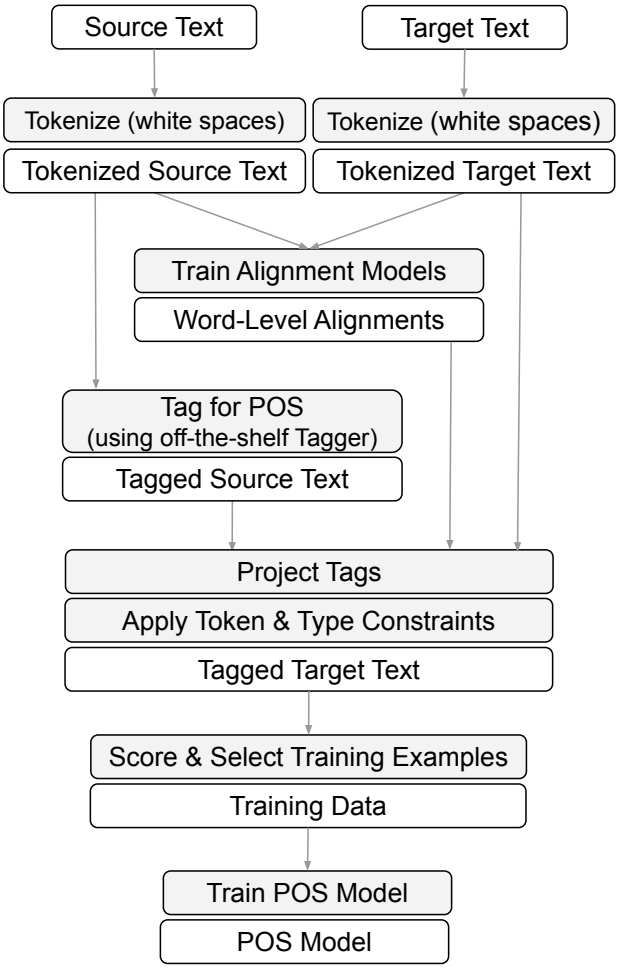
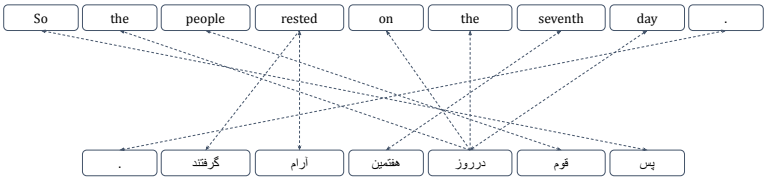


Figure 4.2: The overall pipeline of unsupervised cross-lingual POS tagging via annotation projection

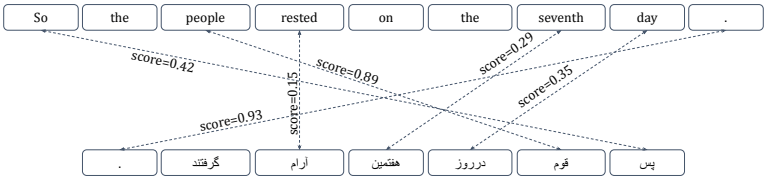
use *MADAMIRA* [3] (Pasha et al., 2014) for Arabic white-space tokenization, for performance gain. However, in order to keep our approach fully unsupervised, we tokenize the target side by applying a large set of language-independent regular expressions that utilizes built-in Python expressions that help recognize punctuation marks and symbols.

Word-Level Alignment Next, we use the sentence-aligned parallel text to train two word-alignment models that align the texts of the source and target sides at the word level in both directions. We experiment with two language-independent unsupervised alignment systems, namely

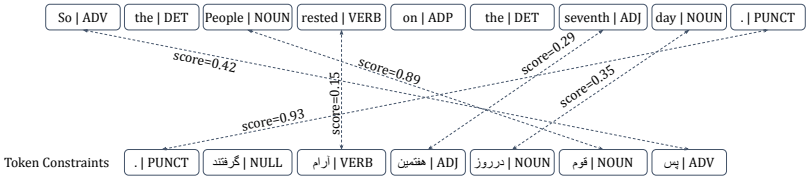
[3] https://camel.abudhabi.nyu.edu/madamira



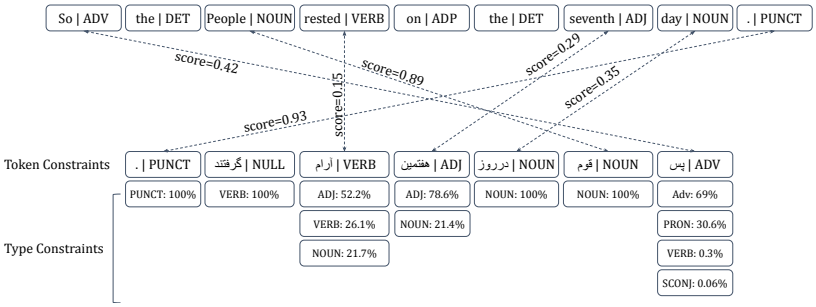
(a) Word-level alignments



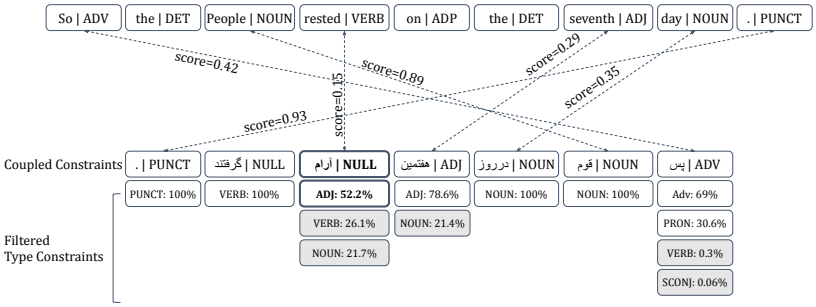
(b) Filtering out one-directional and low-confidence alignments



(c) Tagging the source and projecting the tags (token constraints)



(d) Generating type constraints



(e) Filtering out type constraints of low probabilities and coupling token and type constraints

Figure 4.3: An English-to-Persian example of alignment and projection for verse *EXO 16:30*, *"So the people rested on the seventh day."*. The alignment models are trained on the entire Bible. Persian reads right to left.

GIZA++ [4] (Och and Ney, 2003) and *Fast_Align* [5] (Dyer et al., 2013), to generate the word-level alignments and choose *GIZA++* to align all of our target-source language pairs as it consistently yields better results.

Our aim is to generate one-to-one word-level alignments of high quality to use as the basis for annotation projection. However, word-level alignments suffer from non-precise translations. In addition, there is no one-to-one correspondence between the words across parallel texts, which results in null, one-to-many and many-to-one alignments. Accordingly, we eliminate long sentences of more than 80 tokens and only consider bidirectional word alignments (intersecting source-to-target and target-to-source alignments). Additionally, we exclude the alignment points where the average of the alignment probabilities in the two directions is below some threshold α.

Referring to the English-Persian example in Figure 4.3, the English and Persian Bibles are used to train word-alignment models in both directions. The alignment models are then used to derive the word-level alignments in both directions for each verse (Figure 4.3a), where the one-directional alignments and alignments of low confidence are eliminated (Figure 4.3b). This results in six bidirectional alignments for verse *EXO 16:30*, namely {*So*, پس}, {*people*, قوم}, {*rested*, آرام}, {*seventh*, هفتمین}, {*day*,درروز} and {.,.}.

Tagging the Source Language for POS. Since cross-lingual projection requires a common POS tagset for the underlying languages, we use the universal POS tagset of the Universal-Dependencies (UD) project [6], which consists of 17 universal POS tags, namely ADJ (adjective), ADP (adposition), ADV (adverb), AUX (auxiliary), CCONJ (coordinating conjunction), DET (determiner), INTJ (interjection), NOUN (noun), NUM (number), PART (particle), PRON (pronoun), PROPN (proper noun), PUNCT (punctuation), SCONJ (subordinating conjunction), SYM (symbol), VERB (verb) and X (a placeholder for all other tags). We use *Stanza* to tag the source-side text except in the case of Arabic, for which we apply *MADAMIRA*, for performance gain, after converting the output PTB tags into their universal cognates. However, since *MADAMIRA* was not designed to follow

[4]http://www.statmt.org/moses/giza/*GIZA++*.html
[5]https://github.com/clab/fast_align
[6]https://universaldependencies.org/u/pos

the UD guidelines, we manually correct the mapped Arabic analyses of the most frequent 2,500 POS-lemma pairs by selecting the most likely analysis for each pair.

POS Projection using Token and Type Constraints. In order to project the POS tags from the source side onto the target one, we use token and type constraints based on the mapping induced by the word-level alignments. The idea of using both token and type constraints was first introduced by Täckström et al. (2013). Type constraints define the set of POS tags a word type can receive. In a semi-supervised learning setup, type constraints can be obtained from an annotated corpus (Banko and Moore, 2004) or from a resource that serves as a POS lookup, such as the Wiktionary [7] (Li et al., 2012; Täckström et al., 2013). For the extraction of type constraints in an unsupervised fashion, we follow the approach proposed by Buys and Botha (2016), where we define a tag distribution for each word type on the target side by accumulating the counts of the different POS tags of the source-side tokens that align with the target-side tokens of that word type. The POS tags whose probabilities are equal to or greater than some threshold β then constitute the type constraints of the underlying word type. As for token constraints, each aligned token on the target side gets assigned the POS tag of its corresponding source-side token.

We combine the token and type constraints in a slightly different way from those by Täckström et al. (2013) and Buys and Botha (2016). If a token is not aligned or its token constraint does not exist in the underlying type constraints, the token becomes unconstrained (i.e., receives a null tag). Otherwise, the token constraint is applied. Those applied token constraints then represent the projected tags. Moreover, in contrast to the previous work, we do not use type constraints to impose restrictions while training the POS model as they restrict the performance of our neural architecture.

Referring to the English-Persian example in Figure 4.3, the English text is tagged for POS, and the tags are projected onto the Persian side as token constraints across the bidirectional alignments, where the Persian word گرفتند receives a null assignment as it is not part of a valid alignment (Figure 4.3c). Next, the type constraints are calculated for each word type across the whole Persian corpus (Figure 4.3d). The type constraints of low probabilities are then removed prior to coupling

[7] https://wiktionary.org

the token and type constraints. This results in the removal of the VERB assignment for the Persian word آرام as it is not part of the refined type constraints (Figure 4.3e).

Selection of Training Instances. In a typical supervised-learning setup, adding more training instances helps improve the performance of the system until saturation takes place. On the contrary, adding more training instances that are induced in an unsupervised manner may introduce noise that restricts the quality of the learned model. Accordingly, prior to training a POS model using the projected tags, we score the target sentences based on their "annotation" quality and exclude the ones whose scores are below some threshold γ. We define sentence score as the harmonic mean of its density S_{d} and alignment confidence S_{a}, where S_{d} is the percentage of tokens with projected tags, and S_{a} is the average alignment probability of those tokens.

$$Score(S) = \frac{2(S_{\mathrm{d}} \cdot S_{\mathrm{a}})}{(S_{\mathrm{d}} + S_{\mathrm{a}})}$$

Filtering out sentences of low density and alignment confidence is crucial for training the model. While choosing the sentences with top alignment scores has proved successful in previous research (Duong et al., 2013), we add the density factor as our neural architecture benefits from longer contiguous labeled sequences.

Referring to the English-Persian example in Figure 4.3, the score of the Persian verse is calculated as the harmonic mean of its density and alignment confidence as follows:

$$S_{\mathrm{d}} = \frac{\text{No. of assigned tokens = 5}}{\text{No. of all tokens = 7}} = 0.714$$

$$S_{\mathrm{a}} = \frac{\text{Sum of alignment probabilities}}{\text{No. of assigned tokens}} = \frac{0.42+0.89+0.35+0.29+0.93}{5} = 0.576$$

$$Score(S) = \frac{2(S_{\mathrm{d}} \cdot S_{\mathrm{a}})}{(S_{\mathrm{d}} + S_{\mathrm{a}})} = \frac{2 \cdot 0.714 \cdot 0.576}{0.714+0.576} = 0.638$$

4.2.2 Neural Part-of-Speech Tagging

The architecture of our POS tagger is a bidirectional long short-term memory (BiLSTM) neural network (Hochreiter and Schmidhuber, 1997). BiLSTMs have been widely used for POS tagging

(Huang et al., 2015; Wang et al., 2015; Plank et al., 2016; Ma and Hovy, 2016; Cotterell and Heigold, 2017) and other sequence-labeling tasks, such as named-entity recognition (Lample et al., 2016). The input to our BiLSTM model is labeled sentences, where the labels are automatically generated through alignment and projection (Section 4.2.1), while the word representation is the concatenation of four types of embeddings: pre-trained (*PT*) transformer-based contextualized word embeddings; 2) randomly initialized (*RI*) word embeddings; 3) affix embeddings of 1, 2, 3, and 4 characters; and 4) word-cluster embeddings. Figure 4.4 shows the complete structure of our neural architecture.

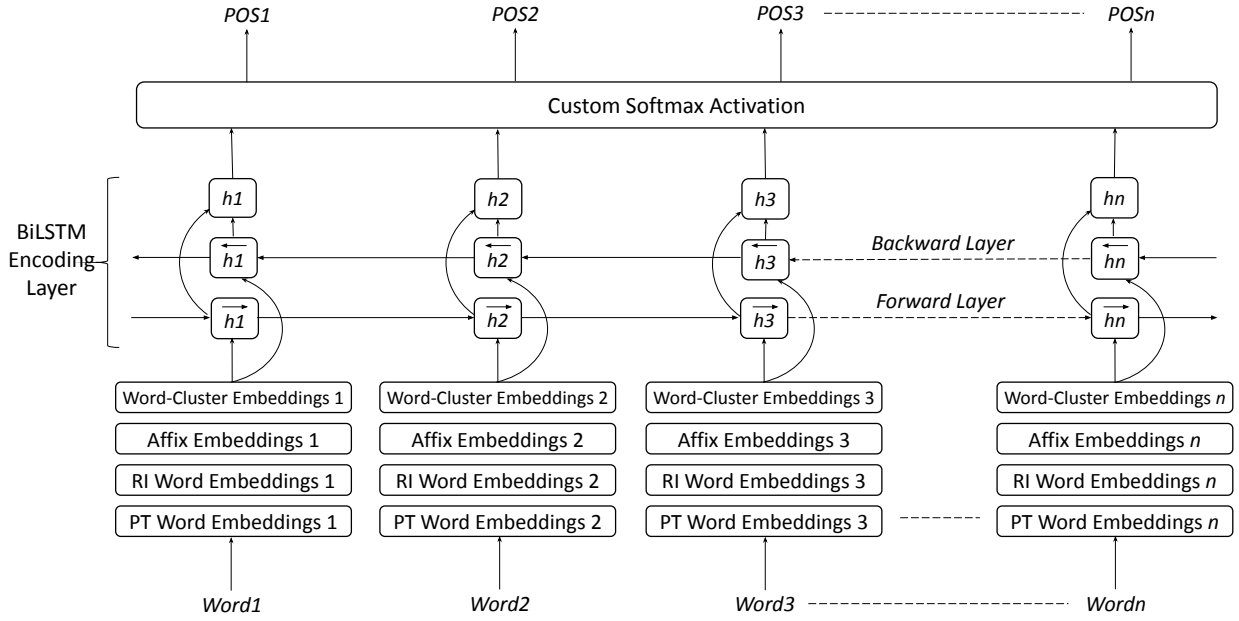


Figure 4.4: The architecture of our BiLSTM neural-network model for POS Tagging. The input annotations are generated through alignment and projection in a fully unsupervised manner. The input layer is composed of the concatenation of four types of embeddings: 1) pre-trained (*PT*) transformer-based contextualized word embeddings; 2) randomly initialized (*RI*) word embeddings; 3) affix embeddings of 1, 2, 3, and 4 characters; and 4) word-cluster embeddings. The model is based on a BiLSTM encoding layer and uses a custom softmax activation that handles null assignments.

Word Embeddings. We use two types of word-embedding features: pre-trained (*PT*)) transformer-based contextualized embeddings and randomly initialized (*RI*) embeddings. For the pre-trained transformer-based contextualized word embeddings, we use the final layer of the multilingual *XLM-RoBERTa* transformer-based language model (*XLM-R*) (Conneau et al., 2019).

XLM-R is a multilingual transformer-based language model that is pre-trained on texts of 100 languages, and its performance is competitive with strong monolingual models when tested for a variety of NLP tasks. However, when applying our neural architecture for a test language that is not represented in the *XLM-R* model, one can consider training a custom monolingual transformer-based language model [8] given the availability of monolingual texts and suitable computational resources (computing power and training time), and thus our architecture is not limited to the languages available in the *XLM-R* model.

It is noteworthy to mention that we obtain better results when using the *XLM-R* embeddings as embedding features as opposed to performing fine-tuning, where the latter is more suitable for sentence-level predictions. Additionally, we examined the use of the multilingual *BERT* model (*mBERT*) (Devlin et al., 2019) instead of *XLM-R*, but this resulted in an average decrease of 0.5% in POS accuracy. We also experimented with the addition of a Conditional-Random-Fields (CRF) layer, but it did not improve the model, which is in line with previous research on POS tagging (Yang et al., 2018; Plank and Agić, 2018). Finally, we use the average of the embedding vectors of the first and last sub-tokens of each word to represent its pre-trained transformer-based contextualized embeddings. This gives us better empirical results than using the embeddings of only the first token or the longest one.

The randomly initialized embeddings are based on the target side of the parallel text and are learned as part of training the model. Coupling both the pre-trained embeddings and the randomly initialized ones is essential when the domain of the training data is different from the one of the pre-trained embeddings, which is the case in our learning setup, where we use the Bible text for training, while the *XLM-R* model is trained on texts from Wikipedia [9] and the CommonCrawl corpus

[8] https://github.com/facebookresearch/XLM
[9] https://wikipedia.org

(See Conneau et al. (2019) for more details).

Affix Embeddings. The use of affix information has proved effective in POS tagging (Ratnaparkhi, 1996; Martins and Kreutzer, 2017). We therefore use randomly initialized prefix and suffix n-gram character embeddings, where n is in $\{1, 2, 3, 4\}$. Our experiments show that affix embeddings are more efficient for POS tagging than character embeddings across the entire words.

Word-Cluster Embeddings The use of word clusters for POS tagging was first proposed by Kupiec (1992), where they classified the words into 400 distinct ambiguity classes for supervised POS tagging. Word clustering has then proved efficient for unsupervised POS tagging (Täckström et al., 2013; Buys and Botha, 2016). In this thesis, we follow Owoputi et al. (2012) by utilizing hierarchical Brown clustering (Brown et al., 1992), which is an HMM-based clustering of a binary merging criterion based on the logarithmic probability of a context under a class-based language model, where the objective is to reduce the loss in the average mutual information (AMI).

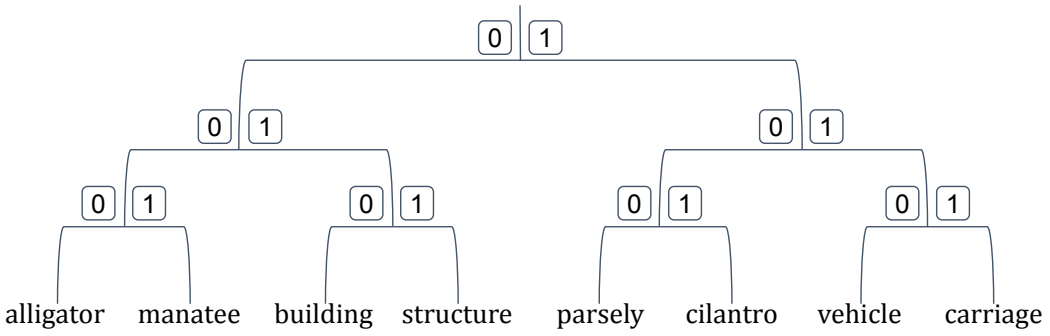
The output of hierarchical Brown clustering is a binary tree of n leaf nodes that represent n word clusters, where each word in the vocabulary belongs to a single leaf cluster. Leaf clusters are recursively grouped into parent ones (interior nodes) until a super cluster of the entire vocabulary is reached (the root).

We produce hierarchical Brown clusters for each target language by applying Percy Liang's implementation of Brown clustering [10] (Liang, 2005) on a monolingual text that is the combination of the Wikipedia and Bible texts of the language. The text is white-space tokenized using the approach described in Section 4.2.1, while the download and clean-up of Wikipedia dumps are handled using the Fairseq library [11].

For each word, we concatenate the main cluster (the binary representation of the corresponding leaf node) with all of its ancestors (the prefixes of the binary representation) to generate the embedding vector that represents the clustering information of the word. This allows us to use the

[10] https://github.com/percyliang/brown-cluster
[11] https://github.com/pytorch/fairseq

hierarchical clustering information and avoid the commitment to a specific granularity level, where high-level clusters may be insufficient, while the lower ones may represent over-clustering.



Word	Cluster Embedding
alligator	[0] | [00] | [000] = [000000]
manatee	[1] | [10] | [100] = [110100]
building	[0] | [01] | [010] = [001010]
structure	[1] | [11] | [110] = [111110]
parsely	[0] | [00] | [001] = [000001]
cilantro	[1] | [10] | [101] = [110101]
vehicle	[0] | [01] | [011] = [001011]
carriage	[1] | [11] | [111] = [111111]

Figure 4.5: An example of a Brown-cluster hierarchy. The word-cluster embeddings are the concatenation of the main leaf clusters with all of their ancestors.

Figure 4.5 shows an example of a Brown-cluster hierarchy of depth two and its corresponding word-cluster embeddings. In our experiments, we use a hierarchy of depth six; that is 128 leaf clusters that represent the whole vocabulary of the underlying language.

Custom Softmax Activation. We use softmax activation on top of the BiLSTM encoding layer for the computation of the output class. However, since some words receive null assignments as a result of missing and rejected alignments and non-overlapping token and type constraints (Section 4.2.1), we set the value of the output neuron corresponding to the null tag to $-\infty$ so that it does not contribute to the calculation of the softmax probabilities and prohibit the model from decoding null values. Moreover, we mask those words of null assignments for the calculation of network loss.

4.3 Languages and Data

We conduct our cross-lingual POS-tagging experiments on six high-source languages and 14 simulated low-resource target ones of diverse typologies. This gives a total of 84 target-source language pairs.

The source languages are chosen to be widely-spoken ones. This is because in a low-resource scenario, available parallel texts are highly likely to include one of them, which is usually the case when translating religious books, movie scripts and user manuals. These languages are English (Indo-European (IE), Germanic), Spanish (IE, Romance), French (IE, Romance), German (IE, Germanic), Russian (IE, Slavic) and Arabic (Afro-Asiatic, Semitic).

We choose 14 simulated low-resource target languages that belong to different language families/genuses and typologies. While some of the target languages are high-resource, we use them in a simulated low-resource setup, where the POS tagging is performed in a fully unsupervised fashion. The target languages are Afrikaans (IE, Germanic), Amharic (Afro-Asiatic, Semitic), Basque (language isolate), Bulgarian (IE, Slavic), Finnish (Uralic, Finnic), Georgian (Kartvelian), Hindi (IE, Hindi), Indonesian (Austronesian, Malayo-Sumbawan), Kazakh (Turkic, Northwestern), Lithuanian (IE, Baltic), Persian (IE, Iranian), Portuguese (IE, Romance), Telugu (Dravidian, South Central) and Turkish (Turkic, Southwestern).

We use the multilingual parallel Bible corpus [12] by Christodouloupoulos and Steedman (2014) as the source of parallel data for all the languages except Georgian and Kazakh [13]. The Bible text is available in full for our source and target languages except Basque, Georgian and Kazakh, in which only the New Testament is available. The limited text available in Basque and its being a language isolate make it an ideal case of cross-lingual learning in a low-resource scenario.

Table 4.1 lists the average number of parallel sentences per target language, across the source languages, (the second column) and the corresponding average number of training sentences after applying the sentence-selection mechanism described in Section 4.2.1 (the third column).

[12] http://christos-c.com/Bible

[13] We collected the biblical texts for Georgian and Kazakh from https://github.com/cysouw/MissingBibleVerses.

Target Language	Parallel sentences	Training sentences
Afrikaans	31,044	23,784
Amharic	30,521	10,045
Basque	7,949	7,225
Bulgarian	31,045	21,600
Finnish	31,000	23,998
Georgian	7,954	7,794
Hindi	31,015	16,105
Indonesian	29,594	9,570
Kazakh	5,873	4,330
Lithuanian	31,083	25,653
Persian	30,965	17,517
Portuguese	31,069	26,751
Telugu	31,085	10,144
Turkish	30,188	16,029
Average	25,742	15,753

Table 4.1: The average number of alignment and training sentences per target language, across the source languages, when using the Bible as the source of parallel data

Indonesian, Telugu and Amharic experience the maximum loss in the number of sentences selected as training instances with relative reductions of 67.7%, 67.4% and 67.1%, respectively, while the average relative reduction across the target languages is 38.8%. It is worth noting that we run the approach on verses as opposed to sentences, which are not equivalent in the rare cases in which a verse contains multiple sentences or a sentence spans multiple verses.

For testing, we use the test datasets of the Universal-Dependencies (UD) project, UD-v2.5 (Zeman et al., 2017) to evaluate our tagging models in terms of POS accuracy. The corpora are *Afrikaans-AfriBooms*, *Amharic-ATT*, *Basque-BDT*, *Bulgarian-BTB*, *Finnish-TDT*, *Hindi-HDTB*, *Indonesian-GSD*, *Kazakh-KTB*, *Lithuanian-ALKSNIS*, *Persian-Seraji*, *Portuguese-Bosque*, *Telugu-MTG* and *Turkish-IMST*. We also report our results on older versions of the UD project in order to compare to the state-of-the-art systems, when needed. One exception is Georgian, for which we developed a small POS-labeled dataset of 100 sentences [14], following the UD-tagging schema, as it is not part of the UD. The sentences are taken from the Modern Georgian and Political texts

[14]https://github.com/rnd2110/unsupervised-cross-lingual-POS-tagging/blob/main/data/KAT-eval.txt

sub-corpora of the Georgian National Corpus [15], and they are hand-tagged and carefully revised by a linguist who specializes in and speaks Georgian as a second language.

Finally, we evaluate our approach for cross-lingual POS tagging via annotation projection versus zero-shot model transfer on Japanese as a case study, where we use the Japanese test set from the CoNLL-2017 shared task (Zeman et al., 2017) for evaluation.

4.4 Evaluation and Analysis

4.4.1 Experimental Settings

The alignment and projection thresholds as well as the hyperparameters of the neural model are manually tuned on Bulgarian, Basque, Finnish and Indonesian when projecting from English using the UD development sets. Table 4.2 lists the search ranges and the final selected values for the alignment and projection thresholds, while Table 4.3 reports on the tuning of the hyperparameters of the neural architecture.

Threshold	Search Domain	Selected Value
Alignment Threshold α	0.1, 0.2, 0.3	0.1
POS-Type Distribution Threshold β	0.1, 0.2, 0.3, 0.4, 0.5	0.3
Sentence-Selection Threshold γ	0.3, 0.4, 0.5, 0.6, 0.7	0.5

Table 4.2: The tuning of the alignment and projection thresholds

Parameter	Search Domain	Selected Value
LSTM Embedding Size	64, 128, 256	128
Randomly-Initialized Embedding Size	64, 128	64
Learning Rate	0.0001 to 0.0005 (steps of 0.00005)	0.0001
Learning-Decay Rate	0, 0.05, 0.1	0.1
Optimizer	SGD, Adam	Adam
Regularization	L2, Dropout	L2 and Dropout
Dropout Rate	0, 0.5, 0.6, 0.7, 0.8	0.7
Number of Epochs	1 to 20 (steps of 1)	12

Table 4.3: The tuning of the neural hyperparameters

[15] http://gnc.gov.ge

We set the alignment threshold α to 0.1 and the threshold γ for the selection of training instances to 0.5. In addition, the POS-type distribution threshold β is set to 0.3, which has proved effective by Banko and Moore (2004) and Buys and Botha (2016).

Our BiLSTM networks are one-layer deep with 128 nodes, while the size of all the randomly initialized word and affix embeddings is 64. We use Adam for optimization (Kingma and Ba, 2014) with a learning rate of 0.0001 and a learning decay rate of 0.1 at each epoch for a total of 12 epochs. To avoid overfitting, we apply L2 regularization and two dropout layers, before and after the BiLSTM encoder, with a dropout rate of 0.7. We also use a cross-entropy loss to assess the performance of the model after each epoch.

Finally, we run the training processes for each target-source language pair for three times and report the average POS accuracy over the three runs.

4.4.2 Overall System Performance

Table 4.4 reports the accuracy of our POS taggers for all the 84 language pairs and the average performance per source and target language. The last column reports the upper-bound supervised performance using *Stanza*. However, the supervised performance is unavailable for Amharic and Georgian due to the unavailability of UD training data and for Kazakh due to the insufficient UD training data.

The overall approach achieves an average POS accuracy of 75.5% across all the language pairs. However, there is a noticeable variance in the performance of the different taggers. One main aspect is that languages that belong to the same family transfer best across each other. For instance, English and German yield the best results for Afrikaans (IE, Germanic), while Spanish and Portuguese are the best performing language pair (IE, Romance), and Russian is the best source for Bulgarian (IE, Slavic). One exception is the case of transferring from Arabic to Amharic (Afro-Asiatic, Semitic). One possible reason is that the Arabic analyzer does not follow the UD guidelines (Section 4.2.1), along with the morphological complexity of Arabic, which also affects the performance of all the taggers that use Arabic as the source.

Target Language	Source for Unsupervised Learning							Supervised (upper Bound)
	English	Spanish	French	German	Russian	Arabic	Average	
Afrikaans	**86.9**	83.1	83.9	84.1	76.4	66.1	80.1	97.9
Amharic	**75.3**	74.6	73.9	75.2	73.3	74.4	74.4	NA
Basque	**67.3**	64.6	65.8	66.7	61.7	55.6	63.6	96.2
Bulgarian	85.6	83.2	83.7	80.7	**87.2**	73.4	82.3	98.5
Finnish	**82.8**	80.9	80.0	82.0	78.6	67.7	78.7	97.1
Georgian	82.8	80.1	80.2	82.5	**83.1**	71.2	80.0	NA
Hindi	**73.9**	72.3	72.6	60.9	66.9	60.1	67.8	97.6
Indonesian	**84.1**	83.5	82.9	81.2	82.4	73.7	81.3	93.7
Kazakh	**73.6**	64.7	67.3	68.9	62.1	63.6	66.7	NA
Lithuanian	80.9	78.2	79.0	78.7	**83.3**	69.8	78.3	93.5
Persian	77.2	**78.1**	76.1	76.5	**78.1**	71.2	76.2	81.1
Portuguese	86.1	**88.7**	86.6	81.2	79.5	70.8	82.2	92.3
Telugu	**80.0**	72.3	73.7	75.6	72.7	64.0	73.1	93.8
Turkish	74.3	72.7	**74.7**	72.8	72.0	67.8	72.4	94.7
Average	**79.3**	76.9	77.2	76.2	75.5	67.8	75.5	94.2

Table 4.4: The POS-tagging performance (accuracy) when using the Bible as the source of parallel data. The best results per target language and per source language on average, across the target languages, is in **bold**. The last column reports the upper-bound supervised performance using *Stanza*.

Since English is the most vital language, where its morphological annotation guidelines were the basis for those of other languages, transferring from English yields the best performance for eight target languages, namely Afrikaans, Amharic, Basque, Finnish, Hindi, Indonesian, Kazakh and Telugu. English also gives the best performance on average with an average relative error reduction of 9.2% over French, the second best on-average performing source language. However, while French only yields the best performance for Turkish, Russian is the best source language for four target languages, namely Bulgarian, Georgian, Lithuanian and Persian. On the other hand, Arabic is the lowest performing source language due to its morphological complexity that involves inflection, fusion and affixation, along with the nature of the analyzer, where it does not follow the UD guidelines. On the other hand, the performance of the target languages is mainly impacted by

the four factors described below.

- **Morphological Complexity**: The best on-average performing target languages are those with the least agglutinative/synthetic nature, compared to the others, namely Bulgarian, Portuguese, Indonesian and Afrikaans, while the remaining 10 target languages involve rich affixation with high ratios of word types to word tokens, which in turn impacts both alignment and projection. This encourages the use of the stem or morphemes as the core unit of abstraction when processing morphologically complex languages (Section 6.2).

- **Similarity to Source**: Since five of the six source languages belong to the Indo-European (IE) language family, the IE target languages, such as Bulgarian, Portuguese and Afrikaans, are among the best performing target languages. In contrast, Basque is a language isolate and experiences the lowest on-average performance.

- **Literary Tradition**: Languages with a strong literary tradition that have the Bible introduced in early ages tend to perform relatively well. In those languages, the gap between the nature of the biblical text and that of modern texts is relatively small, which allows the POS models that are trained on the biblical text to better recognize the text in the evaluation sets. An ideal example is Georgian, a language that has a strong literary tradition and has the Bible translated into as early as the fifth century. In contrast, Persian, Amharic, Telugu, Turkish and Hindi are all languages that did not have the Bible translated into before the 17th century. The five languages rank eighth to 12th, respectively, in terms of their on-average POS performance.

- **Availability of Data**: Kazakh and Basque are the two lowest on-average performing languages, in which we only rely on the New Testament as the source of parallel data for alignment and projection. However, Georgian, where we rely on the New Testament, ranks fifth due to its relatively simpler morphology and strong literary tradition.

4.4.3 Performance on Open-Class Tags

Table 4.5 reports the average precision, recall and F1-score for nouns, verbs and adjectives per target language, across the source languages. For complete results per target-source language pair, see Table 2.1 in Appendix B.

Target Language	Noun			Verb			Adjective		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Afrikaans	84.0	91.3	87.1	67.4	91.1	76.9	80.7	**72.3**	**75.7**
Amharic	68.3	78.5	72.9	82.7	71.1	76.3	31.0	22.7	25.6
Basque	58.7	77.6	66.7	50.0	77.6	60.2	30.0	10.3	14.6
Bulgarian	84.6	**95.8**	**89.7**	80.7	88.7	84.3	79.1	50.1	60.4
Finnish	77.2	88.0	82.0	69.9	82.1	75.2	66.8	48.9	55.3
Georgian	75.4	88.1	80.8	69.7	**95.5**	80.5	**83.9**	55.0	64.9
Hindi	65.3	86.6	74.2	50.4	79.3	60.9	59.0	46.8	51.8
Indonesian	72.5	90.0	80.1	84.1	85.3	84.5	58.3	46.3	51.4
Kazakh	65.6	73.8	68.5	47.5	87.1	60.7	31.3	4.9	7.9
Lithuanian	79.1	93.1	85.2	**84.6**	83.0	83.6	51.9	45.2	46.8
Persian	**86.9**	83.9	85.3	40.9	74.6	52.6	81.1	42.9	55.8
Portuguese	83.4	94.2	88.2	83.1	90.0	**86.3**	69.1	60.8	63.8
Telugu	72.1	60.5	65.7	69.8	92.3	79.3	27.4	14.4	16.4
Turkish	70.9	79.0	74.4	75.5	85.1	79.9	71.3	24.3	35.5
Average	74.6	84.3	78.6	68.3	84.5	74.4	58.6	38.9	44.7

Table 4.5: The average precision, recall and F1-score for nouns, verbs and adjectives per target language, across the source languages, when using the Bible as the source of parallel data. The best result per POS tag and evaluation metric is in **bold**.

The best F1-scores for nouns, verbs and adjectives are achieved in Bulgarian (89.7%), Portuguese (86.3%) and Afrikaans (75.7%), respectively. In fact, the F1-scores for nouns are higher than those for verbs and adjectives on average and in ten target languages, where they exceed 85.0% in Afrikaans, Bulgarian, Lithuanian, Persian and Portuguese, while the F1-scores for verbs are higher than those for nouns and adjectives in Amharic, Indonesian, Telugu and Turkish. On the contrary, the F1-scores for adjectives are lower than those for nouns and verbs on average and across all the target languages. The only exception is Persian, in which the F1-score for verbs is the lowest.

The average recall exceeds the average precision for nouns and verbs by absolute 9.7% and

16.2%, respectively. In the case of nouns, the recall in the Indo-European languages is relatively high, reaching the peak in Bulgarian (95.8%) and scores above 93.0% in Portuguese and Lithuanian. In the case of verbs, the highest recalls are achieved in Georgian (95.5%) and Telugu (92.3%), two non-Indo-European languages. On the contrary, the average recall for adjectives is absolute 19.7% below the average precision, where both are surpassed by those for nouns and verbs.

Target Language	Best Source Language		
	Noun	Verb	Adjective
Afrikaans	German	German	German
Amharic	Arabic	German	English
Basque	English	English	English
Bulgarian	Russian	French	Russian
Finnish	English	German	English
Georgian	German	German	German
Hindi	French	French	German
Indonesian	English	English	English
Kazakh	English	English	English
Lithuanian	Russian	Russian	Russian
Persian	Spanish	Arabic	Spanish
Portuguese	Spanish	French	Spanish
Telugu	English	English	English
Turkish	English	French	English

Table 4.6: The best source language for the detection of nouns, verbs and adjectives per target language when using the Bible as the source of parallel data

Table 4.6 lists the best source language for the detection of nouns, verbs and adjectives per target language. In seven target languages, the three tags are best detected by the same source language, while in four target languages, the best source language for the detection of the nominal tags (nouns and adjectives) differs from the best source language for the detection of verbs. On another hand, Amharic is the only target language in which the three tags are best detected by three different source languages, where Arabic is the best source for nouns. Arabic is also the best source language for tagging verbs in Persian. However, while German yields the best performance on the three tags in both Afrikaans and Georgian, it does not result in the best overall performance in either language (Table 4.4) as it is less efficient at the projection of closed-class tags. As seen, different source

languages can be efficient at learning different sets of tags. This encourages developing models that utilize multiple source languages and learn when to trust a source for a specific tag (Section 5.2.1).

Target Language	Noun			Verb			Adjective		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
English	**78.8**	85.5	**81.7**	**74.7**	80.6	76.9	62.0	**48.8**	**52.9**
Spanish	77.9	81.7	79.5	67.7	85.7	74.4	**64.4**	42.7	48.8
French	76.5	83.9	79.9	68.2	86.5	75.6	61.2	38.6	45.9
German	76.4	84.6	80.0	74.0	82.5	**77.2**	58.2	40.5	46.3
Russian	76.5	81.1	78.4	61.7	**90.6**	71.9	54.6	40.8	44.9
Arabic	61.2	**89.0**	72.1	63.6	81.1	70.3	51.4	22.0	29.4
Average	74.6	84.3	78.6	68.3	84.5	74.4	58.6	38.9	44.7

Table 4.7: The average precision, recall and F1-score for nouns, verbs and adjectives per source language, across the target languages, when using the Bible as the source of parallel data. The best result per POS tag and evaluation metric is in **bold**.

Table 4.7 reports the precision, recall and F1-score for nouns, verbs and adjectives per source language, across the target languages. Projecting from English achieves the best average F1-scores for nouns and adjectives (81.7% and 52.9%, respectively), while projecting from German achieves the best average F1-score for verbs (77.2%). Projecting from German is also effective for nouns and adjectives, ranking second and third, respectively, which indicates that German is an efficient source for open-class tags, achieving the second best overall performance next to English. In contrast, projecting from Arabic yields the lowest F1-scores for the three tags.

4.4.4 Ablation Setups

We examine two ablation setups:

- **No_XLM**: In this setup, the transformer-based contextualized word embeddings, *XLM-R*, are excluded. This setup emulates a situation in which the target language is not represented in the *XLM-R* model (nor any equivalent multilingual model) along with the lack of the computational resources needed to train an equivalent monolingual model.

- **No_MONO**: In this setup, both the transformer-based contextualized word embeddings, *XLM-R*, and the word-cluster embeddings are excluded. This setup emulates a situation in which the accessible monolingual data in the target language is not sufficient to learn rich representations of the language.

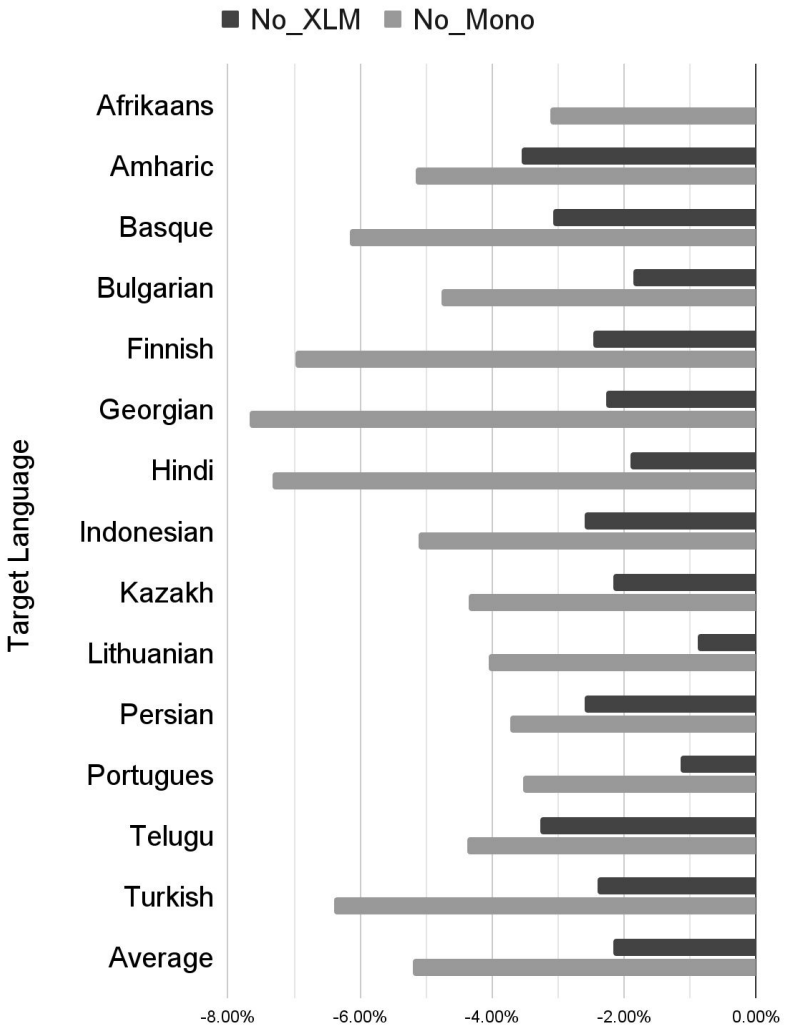


Figure 4.6: The average drop in POS accuracy per target language, across the source languages, in the No_XLM (dark gray) and No_Mono (light gray) ablation setups when using the Bible as the source of parallel data

We evaluate the No_XLM and No_Mono setups on the 84 language pairs using a doubled

learning rate (from 0.0001 to 0.0002) as the complexity decreases, and we report the average drop in POS accuracy per target language, across the source languages, in Figure 4.6.

The average POS accuracy across all the target languages decreases by absolute 2.2% and 5.2% in the No_XLM and No_Mono setups, respectively. The impact of eliminating the *XLM-R* embeddings is mostly noticeable in Basque, Telugu and Amharic, where they experience the highest performance drop of more than 3.0%. On the contrary, the performance for Afrikaans remains unchanged as it is under-represented in the multilingual representations. On the other hand, the ablation of the monolingual models affects Finnish, Hindi and Georgian the most, causing absolute reductions in POS accuracy of at least 7.0% in the No_Mono setup. In addition, the use of Brown clusters benefits Georgian and Hindi the most, yielding an absolute performance increase of 5.4%.

The performance drop in the No_Mono setup highlights the importance of monolingual data, which is key to the competitive performance of our POS taggers when compared to other state-of-the-art systems (Section 4.4.5). However, the on-average performance of the system in the absence of only the *XLM-R* embeddings drops by only 2.2%, which provides a relatively good compromise when one lacks adequate computational resources to train such rich representations.

4.4.5 Comparison to State-of-the-Art

Next, we show that our system outperforms the state-of-the-art unsupervised and semi-supervised cross-lingual POS taggers, where the refinement of the annotations and the rich word representation in the neural architecture are more efficient than using larger and/or domain-appropriate parallel data or some labeled data.

We first compare our approach to the state-of-the-art unsupervised system by Buys and Botha (2016), denoted by *BUYS*. *BUYS* performs fully unsupervised cross-lingual POS tagging using a neural model that is based on Wsabie (Weston et al., 2011), a shallow neural network that focuses on optimizing precision at the top of a ranked list of tags. In *BUYS*, the tags are learned through alignment and projection, where each token is assigned either a tag that represents the token constraint or a ranked list of tags that represents the type constraints when the token constraint

is missing due to a null alignment. In contrast, we use type constraints to eliminate those token constraints of low probabilities across the target side, and we do not use type constraints to impose restrictions in our neural model. Another main difference is that *BUYS* relies on large parallel data, Europarl [16] (Koehn, 2005), whose domain is similar to that of the UD test sets as the source for alignment and projection. This makes the system not ideal for truly low-resource scenarios, as opposed to our low-resource setup using the Bible.

We evaluate the performance of our system versus *BUYS* on the shared target-source language pairs, namely {Bulgarian, English}, {Finnish, English}, {Portuguese, English} and {Portuguese, Spanish}, and we report the results on the test sets of UD-v1.2 (as in the evaluation by Buys and Botha (2016)) in Table 4.8. Despite the use of smaller and out-of-domain parallel data, our approach outperforms *BUYS* on all the language pairs with an average relative error reduction of 8.5%

Target	Source	*BUYS*	Our System
Bulgarian	English	81.8	**83.3**
Finnish	English	77.1	**80.4**
Portuguese	English	84.3	**84.6**
Portuguese	Spanish	88.0	**89.1**

Table 4.8: Comparison to *BUYS*, an unsupervised system for cross-lingual POS tagging, in terms of POS accuracy. The best result per language pair is in **bold**.

Next, we compare our approach to the state-of-the-art semi-supervised system by Cotterell and Heigold (2017), denoted by *CTRL*. *CTRL* is a character-level RNN tagger that jointly learns the morphological tags of a high-resource language and the target one. It has two experimental setups that utilize 100 and 1,000 manually annotated target tokens, denoted by D100 and D1000, respectively.

We evaluate the performance of our system versus *CTRL* on the two shared target-source language pairs, namely {Bulgarian, Russian} and {Portuguese, Spanish}, and we report the results using the test sets of UD-v2.0 (as in the evaluation by Cotterell and Heigold (2017)) in Table 4.9. Despite the use of no manually labeled data, our approach outperforms the D100 and D1000 setups

[16] http://statmt.org/europarl

except the D1000 setup with the {Portuguese, Spanish} language pair, where the performance of our system is only 0.2% behind that of *CTRL*. In addition, our approach achieves average relative error reductions of 48.5% and 19.0% over the D100 and D1000 setups, respectively.

Target	Source	*CTRL* D100	*CTRL* D1000	Our System
Bulgarian	Russian	68.8	83.1	**87.2**
Portuguese	Spanish	81.8	**88.9**	88.7

Table 4.9: Comparison to *CTRL*, a semi-supervised system for POS tagging, in terms of POS accuracy. The best result per language pair is in **bold**.

4.5 Annotation Projection vs. Supervised Learning

Next, we compare the best annotation-projection performance of each target language to the corresponding upper-bound supervised performance by *Stanza* in Table 4.4. On average and in eight target languages (out of the eleven languages for which the supervised performance is known), namely Afrikaans, Bulgarian, Finnish, Indonesian, Lithuanian, Persian, Portuguese and Telugu, the unsupervised taggers successfully predict at least as many as 85.0% of the correct decisions made by their corresponding supervised ones, where the percentage exceeds 95.0% in the cases of Persian and Portuguese.

For a given language for which a POS tagger is needed, one should consider the unsupervised path before taking the decision to build a supervised tagger, especially with the cost and time associated with labeling data for POS, which might not be even possible for some low-resource languages. The impact of the small performance gaps between the unsupervised and supervised approaches in some languages might be tolerable when utilizing the POS tags within a downstream task. Moreover, the performance of the unsupervised taggers might be adequate for the tags of interest.

One relevant question is "How many labeled words are needed in order to build a supervised POS tagger that approximates the performance of the unsupervised one?" To answer this question, we use the UD-v2.5 training sets to train supervised taggers of different word-based sizes that are

divisible by 100 in order to find the size of the training data needed to obtain the performance of the annotation-projection approach. We do this for each target language for which the training data is available and adequate, and we report the results in Table 4.10 with respect to the best unsupervised performance in Table 4.4.

Language	Annotation Size	POS Accuracy %
Afrikaans	4,100	86.9
Basque	1,200	67.3
Bulgarian	2,400	87.2
Finnish	5,200	82.8
Hindi	1,900	73.9
Indonesian	2,900	84.1
Lithuanian	6,600	83.3
Persian	2,000	78.1
Portuguese	6,900	88.7
Telugu	1,200	80.0
Turkish	2,600	74.7
Average	3,364	80.6

Table 4.10: The training size (in words) of the supervised tagger that approximates the performance of the best unsupervised setup per target language

On average, it is needed to annotate 3,364 words in order to develop a supervised tagger that yields the unsupervised performance, where the training sizes range from 1,200 words, in Basque and Telugu, to 6,900 words, in Portuguese. That is, one can avoid the cost and time needed to label 6,900 words for POS in Portuguese, for instance, when the equivalent unsupervised performance is sufficient.

4.6 Annotation Projection vs. Zero-Shot Model Transfer

One approach to zero-shot cross-lingual POS tagging is to apply a tagging model that was trained for a related language on the target one. Pires et al. (2019) widely investigate zero-shot model transfer for POS tagging and named-entity recognition by fine-tuning the multilingual *BERT* language model (*mBERT*) on some language and applying the fine-tuned model on another. While the approach does not require parallel data between the source and target sides, the pre-trained models do not

generalize well across languages of different typologies.

We compare our annotation-projection approach to the state-of-the-art zero-shot model-transfer approach by Pires et al. (2019), denoted by *PIRES*, for English and Japanese — different language families and typologies. We again use the Bible from the multilingual parallel Bible corpus by Christodouloupoulos and Steedman (2014) as the source of parallel data for alignment and projection. However, we utilize *mBERT* instead of *XLM-R* and train our model for only three epochs in order to replicate the experimental settings by Pires et al. (2019) [17]. As shown in Table 4.11, our approach outperforms zero-shot model transfer with a relative error reduction of 31.6% when evaluated on the Japanese test set from the CoNLL-2017 shared task (Zeman et al., 2017). This result indicates that annotation projection might be less sensitive to the relatedness between the source and target languages than zero-shot model transfer, which is in line with the results in Table 4.4, and thus can better generalize across a variety of languages of different typologies.

Target	Source	*PIRES*	Our System
Japanese	English	49.4	**65.4**

Table 4.11: Comparison to *PIRES*, an approach for zero-shot model transfer via fine-tuning, in terms of POS accuracy

In order to confirm the conclusion above, we study the POS-tagging performance across language pairs of similar and different typological features, where we consider two types of features: 1) Subject-Verb-Object order (SVO vs. SOV); and 2) Adjective-Noun order (AN vs. NA). For the classification of our source and target languages= based on the WALS database [18] (Dryer and Haspelmath, 2013). See Table 4.12.

Tables 4.13 and 4.14 report the macro-average POS accuracies when transferring across language pairs with respect to their Subject-Object-Verb order and Adjective-Noun order, respectively, where we compare annotation projection to zero-shot model transfer (*PIRES*) [19]. The last column indicates

[17] We assume that white-space tokenization is accessible for Japanese, and so do Pires et al. (2019). We use *Stanza* to obtain the white-space tokenized text.

[18] https://wals.info

[19] Strictly speaking, the numbers are not comparable as the languages are different. However, they provide insights

Language	Subject-Verb-Object Order	Adjective-Noun Order
Afrikaans	SVO/VSO	AN
Amharic	SOV	AN
Arabic	VSO	NA
Basque	SOV	NA
Bulgarian	SVO	AN
English	SVO	AN
Finnish	SVO	AN
French	SVO	NA
Georgian	SOV	AN
German	SOV/SVO	AN
Hindi	SOV	AN
Indonesian	SVO	NA
Kazakh	SOV	AN
Lithuanian	SVO	AN
Persian	SOV	NA
Portuguese	SVO	NA
Russian	SVO	AN
Spanish	SVO	NA
Telugu	SOV	AN
Turkish	SOV	AN

Table 4.12: The Subject-Verb-Object order and Adjective-Noun order of our source and target languages

the impact of typological similarity, which is calculated as the relative error reduction due to transferring across languages of similar typological features as compared to transferring across languages of different typological features, e.g., the error reduction of transferring from an SOV language to an SOV one, {SOV, SOV}, as compared to transferring from an SOV language to an SVO one, {SOV, SVO}.

In *PIRES*, the best performance is always achieved when transferring across languages of similar typological features, {SVO, SVO}, {SOV, SOV}, {AN, AN} and {NA, NA}, which is not the case in our approach, where the performance of transferring from SVO languages is comparable to that of SOV sources, while AN and NA sources result in similar performance patterns regardless of the type of the target. Moreover, the impact of typological similarity in *PIRES* is consistently higher than

on how the two approaches perform across typologically diverse languages.

PIRES	SVO	SOV	Impact of Typological Similarity
SVO	81.6	66.5	45.1
SOV	64.0	64.2	0.6
Our System	SVO	SOV	Impact of Typological Similarity
SVO	82.5	72.9	35.4
SOV	81.3	72.4	-47.6

Table 4.13: The macro-average POS accuracies when transferring across SVO and SOV languages. Rows = sources, columns = targets. Impact of Typological Similarity refers to the relative error reduction due to transferring across languages of similar typological features.

PIRES	AN	NA	Impact of Typological Similarity
AN	73.3	70.9	8.2
NA	75.1	79.6	18.1
Our System	AN	NA	Impact of Typological Similarity
AN	77.1	76.8	1.3
NA	73.6	74.8	4.5

Table 4.14: The macro-average POS accuracies when transferring across AN and NA languages. Rows = sources, columns = targets. Impact of Typological Similarity refers to the relative error reduction due to transferring across languages of similar typological features.

the one in our approach across all the compared pairs. This means that annotation projection has a better ability to transfer across languages of diverse typologies than zero-shot model transfer. This can be explained since in the annotation-projection approach, the characteristics of the source only contribute to the alignment and projection phases, while training the POS model is fully conducted in the target space after eliminating low-quality projections. On the contrary, training the model in the zero-short model-transfer approach is fully performed in the source space, making it more difficult to generalize well across unrelated languages.

4.7 Conclusion

In this chapter, we focused on fully unsupervised cross-lingual POS tagging via annotation projection in truly low-resource scenarios, where we use the Bible as the source of parallel data for alignment

and projection as it is available for a large number of languages and meets our low-resource assumptions: small in size and out-of-domain with respect to the evaluation sets.

In order to overcome the annotation-projection challenges that arise from bad translation, erroneous alignments and translation phenomena that affect the quality of word-alignment models, we standardized the process of POS tagging via annotation projection by exploiting and expanding the best practices in the literature as we aim at producing reliable annotations towards efficient POS models. In addition, we designed a powerful BiLSTM neural architecture that uses transformer-based contextualized word embeddings and combines word embeddings, affix embeddings and word-cluster embeddings, along with special handling for null assignments due to missing and rejected alignments and non-overlapping token and type constraints.

We evaluated our approach on six source languages and 14 typologically diverse target languages, for a total of 84 target-source language pairs. Our system achieves an average POS accuracy of 75.5% across all the language pairs, where languages that belong to the same family/genus transfer best across each other. We also demonstrated the efficiency of our approach in the tagging of open-class words, where the average F1-scores for detecting nouns and verbs are 78.6% and 74.4%, respectively.

We showed that our approach outperforms the state-of-the-art unsupervised system by Buys and Botha (2016) despite the use of smaller and out-of-domain parallel data, achieving an average relative error reduction of 8.5%. Our approach also outperforms the state-of-the-art semi-supervised system by Cotterell and Heigold (2017) despite the absence of manually labeled data, achieving an average error reduction of 19.0%.

We conducted ablation setups in which we 1) eliminate the use of transformer-based contextualized word embeddings; and 2) assume limited access to monolingual data, i.e., eliminate both the transformer-based contextualized word embeddings and the word-cluster embeddings. In the two ablation setups, our system still performs relatively well with average performance drops of only 2.2% and 5.2%, respectively.

We also demonstrated that unsupervised cross-lingual learning via annotation projection might

be an alternative to supervised learning, where in eight target languages, our unsupervised taggers successfully predict at least as many as 85.0% of the correct decisions made by their corresponding supervised ones. We also showed that it is required to annotate 3,364 words, on average, in order to build supervised taggers that are comparable to the unsupervised ones.

Finally, we showed that annotation projection transfers better than zero-shot model transfer across languages of diverse typologies and thus is less sensitive to typological dissimilarities. Our approach results in a relative error reduction of 31.6% over zero-shot model transfer in a case study for transferring from English to Japanese.

Chapter 5

Unsupervised Multi-Source Cross-Lingual Part-of-Speech Tagging

5.1 Overview

The availability of parallel corpora that involve multiple languages encourages the use of multiple source languages for cross-lingual POS tagging via annotation projection (Agić et al., 2015; Agić et al., 2016; Plank and Agić, 2018). Examples of such parallel corpora include, but are not limited to, the Bible (Mayer and Cysouw, 2014) (484 languages of complete Bible translations and 2,551 languages of partial translations), the Watchtower Corpus (WTC) [1] (137 languages), the Book of Mormon ($\sim$ 100 languages), Harry Potter (and the Philosopher's Stone) (Rowling, 1997) (73 languages), Dianetic: The Modern Science of Mental Health (Hubbard and Sherr, 2007) (64 languages), Europarl [2] (Koehn, 2005) (21 languages), Hansards [3] and the ODS UN dataset [4]. Translations that involve multiple languages can also be available in other materials such as movie scripts, medical prescriptions and user manuals. However, we choose the Bible as the source of parallel data for alignment and projection as it meets our low-resource assumptions: small in size and out-of-domain with respect to the evaluation sets.

Our contribution is twofold:

- We investigate the use of multiple source languages to induce the tags on the target side. We introduce two multi-source approaches: 1) multi-source projection, where we combine the tags projected from multiple source languages onto the target side prior to training the POS model (Section 5.2.1); and 2) multi-source decoding, where we combine the tags produced by

[1] https://www.jw.org/en/online-help/watchtower-library
[2] http://statmt.org/europarl
[3] https://catalog.ldc.upenn.edu
[4] https://documents.un.org/prod/ods.nsf/home.xsp

multiple single-source models to tag a given text in the target language (Section 5.2.2). Our multi-source setups are based on either weighted maximum voting or Bayesian inference that constructs confusion matrices to learn what sources to rely on for specific sets of tags. We also conduct weighted Bayesian inference, in which we combine both mechanisms in hybrid setups. This makes a total of eight multi-source setups.

- We conduct extensive evaluation and analysis on the 14 target languages we used for evaluating our single-source setups (Sections 5.3 and 5.4).

We show improvements over the single-source setups on average and in 10 target languages. In addition, we demonstrate significant improvements over previous work that relies on multi-source setups, both unsupervised and semi-supervised. We also study ablation setups and show that our multi-source approaches can compensate for some of the performance drop due to the lack of adequate computational resources and/or monolingual data (Section 5.4).

Finally, we show that our multi-source setups are able to predict at least as many as 90.0% of the correct decisions made by the corresponding supervised taggers in four target languages. In addition, we show that more annotations are needed to build equivalent supervised taggers than in the case of using single-source setups (Section 5.5).

This chapter contains our work on the multi-source approaches for unsupervised cross-lingual POS tagging for truly low-resource scenarios (Eskander et al., 2020b) and extends it by providing additional multi-source setups, where we define different types of weights for weighted maximum voting and introduce Bayesian inference to understand the reliability of each source with respect to the POS tags. We also expand our evaluation and analysis to better understand the behavior of the multi-source approaches.

5.2 Methods

We divide our multi-source approaches into two groups:

1. **Multi-Source Projection**: In this approach, we combine projected tags that correspond to multiple source languages after coupling the token and type constraints and prior to training the POS model (Section 5.2.1).

2. **Multi-Source Decoding**: In this approach, we combine the tagging outputs of multiple single-source models applied on some text in the target language (Section 5.2.2).

5.2.1 Multi-Source Projection

As detailed in Section 4.2.1, POS tagging via annotation projection relies on parallel data between a source language and the target one, where the source gets tagged using an accessible tagger, and the tags are projected from the source onto the target across word-level alignments. The projected tags then form token constraints, while the tag distribution of each target word type defines type constraints. The type constraints of low probabilities are then filtered out, while those token constraints that do not match the corresponding type constraints are rejected. The qualifying token constraints then form the annotations needed to train a POS model for the target language.

In this approach, inspired by the work by Agić et al. (2015) and Agić et al. (2016), we rely on projecting the annotations from multiple source languages by utilizing parallel corpora that involve more than two languages, including the target one, to derive multiple POS assignments on the target side. First, both alignment and projection, along with coupling token and type constraints, are conducted between each source language and the target one, separately. This results in multiple POS assignments on the target side, where a token might receive one or more POS tags (including null). We then apply a voting mechanism to select the most likely correct POS tag for each token. Finally, we select the high-scoring sentences as training instances to build the POS model. The process of multi-source projection is illustrated in Figure 5.1.

We develop three voting mechanisms for the selection of the final POS assignment: 1) weighted maximum voting (Section 5.2.1.1); 2) Bayesian inference (Section 5.2.1.2); and 3) Weighted Bayesian inference (Section 5.2.1.3). We describe the three mechanisms below.

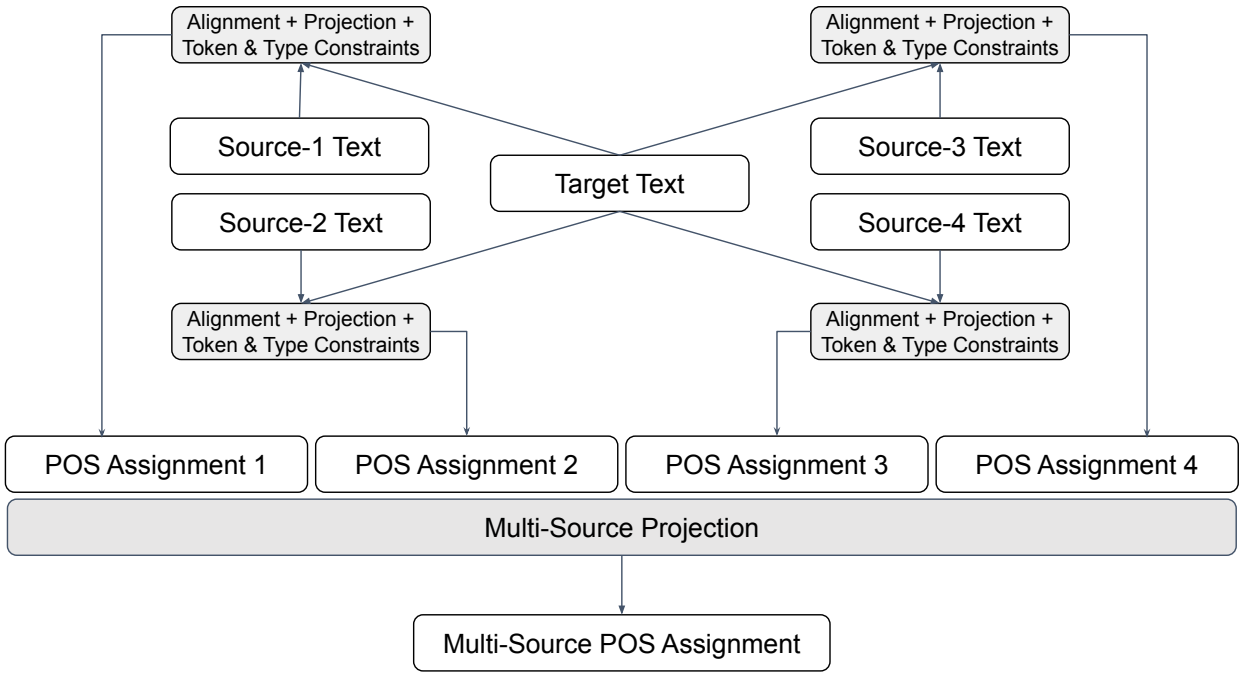


Figure 5.1: The pipeline of multi-source projection (assuming four source languages)

5.2.1.1 Weighted Maximum Voting

For each target token, we assign the projected tag that receives the maximum voting across the source languages weighted by the alignment probability that corresponds to the underlying {token, source language} pair. We denote this voting setup by MP_{wmv}.

Let the POS assignment of a specific {token, source language} pair be y_{ij}, where $i \in [1, N]$ indexes a token, and $j \in [1, H]$ indexes a source language. And let the alignment probability that results in the POS assignment y_{ij} be w_{ij}. The final POS assignment of token i, $\hat{y}_i$, is then defined as:

$$\hat{y}_i = \arg\max_y \sum_{j,\, y_{ij}=y} w_{ij}$$

5.2.1.2 Bayesian Inference

As discussed in Section 4.4.3, different source languages can be efficient at learning different sets of POS tags. Accordingly, one way to improve multi-source voting is to learn when to trust a tagging source for a specific tag. The idea is that good sources tend to have identical predictions across

the corpus, while less reliable ones have divergent predictions. This can be modeled as confusion matrices, one per source, where an unsupervised generative model can learn a proper distribution for the selection of the best source(s) per tag.

We exploit the Bayesian-inference method proposed by Rahimi et al. (2019) for the prediction of named-entity labels based on the outcomes of several named-entity recognition models. We apply the same method for the selection of POS tags given the projection outcomes of multiple source languages, which is a simpler task as aggregated projected labels of named entities might result in invalid schemes, which motivates projecting the labels at the entity level (multiple-token processing). The algorithm is based on a fully unsupervised probabilistic graphical model that is inspired by Kim and Ghahramani (2012). We denote this voting setup by MP_{bys}.

Let the POS assignment of a specific {token, source language} pair be y_{ij}, where $i \in [1, N]$ indexes a token, and $j \in [1, H]$ indexes a source language. The generative model assumes a tag assignment $z_{\mathrm{i}} \in [1, K]$ that is corrupted by the assignment of each source language, y_{ij}. The corruption process is described as follows:

$$P(y_{\mathrm{ij}} = l | z_{\mathrm{i}} = k, V^{(j)}) = {V_{\mathrm{kl}}}^{(j)}$$

where $V^{(j)} \in R_{\mathrm{k\,k}}$ is the confusion matrix that corresponds to source language j. The confusion matrices are drawn from independent Dirichlet priors, with a parameter $\alpha = 1$, and the tags are controlled by a Dirichlet prior π that is drawn from an uninformative Dirichlet distribution, where all the parameters are set to values of one.

When multiple source languages result in identical assignments, k, for some token, this can be explained by assigning $z_{\mathrm{i}} = k$ and assigning high weights to ${V_{\mathrm{k\,k}}}^{(j)}$ in the corresponding confusion matrices. The other less reliable source languages will result in divergent assignments that are likely to be in disagreement or biased towards some other tag, where the model uses the off-diagonal elements to explain such assignments. By aggregating over all the tokens, the model can then differentiate between the reliable source languages (those with high ${V_{\mathrm{k\,k}}}^{(j)}$ values) and the less reliable ones (those with high ${V_{\mathrm{k\,l}}}^{(j)}$ values, where $k \neq l$) for each tag, i.e., the reliability of a source language is calculated with respect to a specific tag.

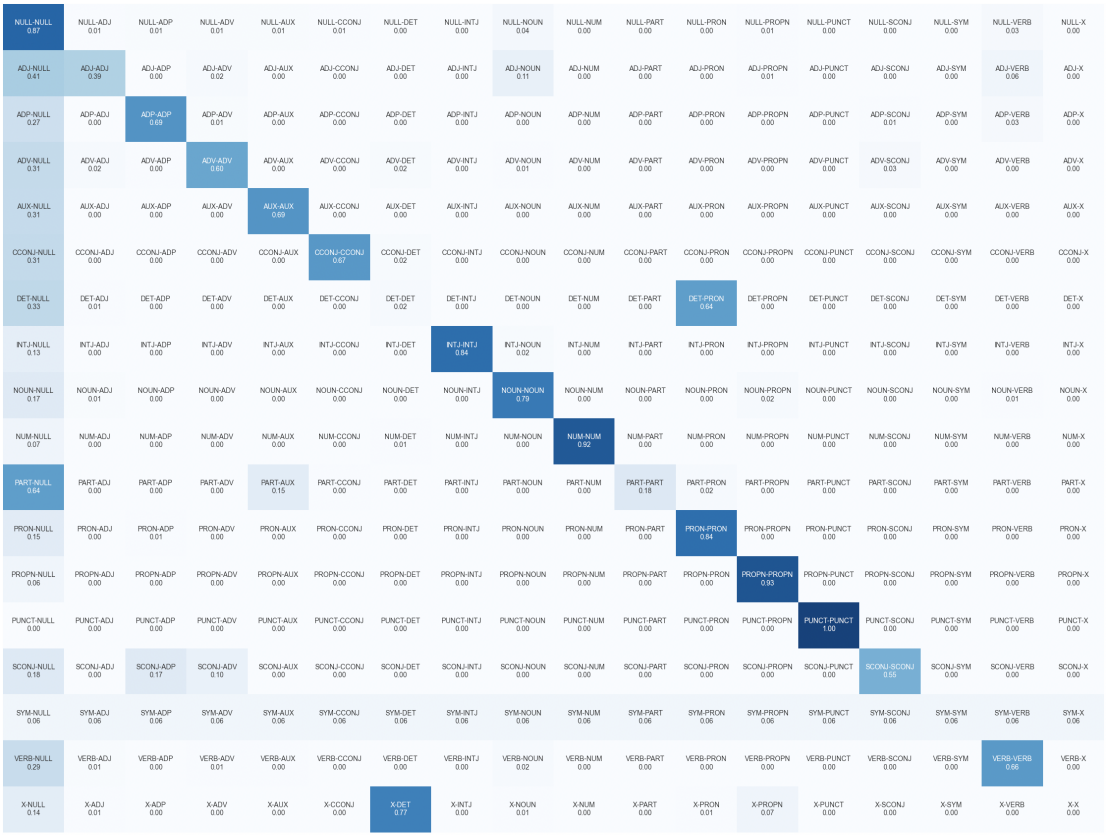


Figure 5.2: A confusion matrix for the projection from English to Finnish

Figure 5.2 visualizes the confusion matrix when projecting from English to Finnish using the Bible as the source of parallel data. The diagonal cells in dark blue refer to reliable assignments when projecting from English, which are those that are mostly in agreement with the assignments of other sources. For example, the NUM, PROPN and PUNCT assignments are the most reliable, followed by those of INTERJ, NOUN and PRON. On the other hand, the DET, SYM and X assignments are the least reliable (the lightest shade on the diagonal), where the off-diagonal cell DET-PRON in relatively dark blue indicates that DET is likely to be confused with PRON, while the off-diagonal cell X-DET in relatively dark blue indicates that X is likely to be confused with DET.

Mean-field variational (Jordan, 1998) is used for inference, in which a fully variational distribution $q(Z, V, \pi) = q(Z)q(V)q(\pi)$ is learned to optimize the evidence lower bound (ELBO). The inference is initialized as the number of votes per POS tag each token receives, which results in a bias towards maximum voting. An iterative learning process with the update rules below then takes

place until convergence.

$$\mathbb{E}_{\mathrm{q}} \log \pi_{\mathrm{k}} = \psi(\beta + \sum\nolimits_{\mathrm{i}} q(z_{\mathrm{i}} = k) - \psi(K\beta + N)$$

$$\mathbb{E}_{\mathrm{q}} log V_{\mathrm{kl}}{}^{(j)} = \psi(\alpha + \sum\nolimits_{\mathrm{i}} q(z_{\mathrm{i}} = k)) - \psi(K\alpha + \sum\nolimits_{\mathrm{i}} q(z_{\mathrm{i}} = k))$$

$$q(z_{\mathrm{i}} = k) \propto exp\{\mathbb{E}_{\mathrm{q}} \log \pi_{\mathrm{k}} + \sum\nolimits_{\mathrm{j}} \mathbb{E}_{\mathrm{q}} log V_{\mathrm{ky_{ij}}}{}^{(j)}\}$$

where ψ is defined as the logarithmic derivative of the gamma function, digamma.

The final POS assignment of each token is then calculated based on $q(Z)$ using the maximum-a-posteriori tag as follows:

$$\hat{\mathrm{z}}_{\mathrm{i}} = \arg\max\nolimits_{\mathrm{z}} q(z_{\mathrm{i}} = z)$$

5.2.1.3 Weighted Bayesian Inference

We combine the Bayesian-inference setup MP_{bys} with the weighted maximum-voting setup MP_{wmv}, described in Section 5.2.1.1, in one hybrid setup, denoted by MP_{wbys}.

Let the alignment probability that results in the POS assignment y_{ij} be w_{ij}. We define the inference as a fully variational distribution $q(Z, V, W, \pi)$. The inference is initialized as the weighted number of votes per POS tag each token receives, where we sum the probabilities of the alignments resulting in the corresponding {token, tag} pair across the source languages, $w_{\mathrm{iz}} = \sum_{\mathrm{j,y_{ij}\,=\,z}} w_{\mathrm{ij}}$.

The final POS assignment of each token is then calculated using the weighted maximum-a-posteriori tag as follows:

$$\hat{\mathrm{z}}_{\mathrm{i}} = \arg\max\nolimits_{\mathrm{z}} q(z_{\mathrm{i}} = z) \cdot w_{\mathrm{iz}}$$

5.2.1.4 Example of Multi-Source Projection

Table 5.1 shows two examples of single-source and multi-source projection for Finnish (upper part) and Portuguese (lower part) when using the Bible as the source of parallel data. The example corresponds to verse *JOH 10:42* in the Bible.

Source Language / Multi-Source Setup	Finnish Verse *JOH 10:42*					
	Ja (*And*) CCONJ	*monet* (*many*) ADJ	*siellä* (*there*) ADV	*uskoivat* (*believed*) VERB	*häneen* (*in him*) PRON	. (.) PUNCT
English	_	ADJ	ADV	VERB	_	PUNCT
Spanish	CCONJ	PRON	ADV	VERB	_	PUNCT
French	CCONJ	_	_	VERB	_	PUNCT
German	CCONJ	PRON	_	VERB	_	PUNCT
Russian	CCONJ	NUM	ADV	VERB	PRON	PUNCT
Arabic	_	_	ADV	VERB	_	PUNCT
MP_{wmv}	CCONJ	PRON	ADV	VERB	PRON	PUNCT
MP_{bys}	CCONJ	ADJ	ADV	VERB	PRON	PUNCT
MP_{wbys}	CCONJ	PRON	ADV	VERB	PRON	PUNCT

Source Language / Multi-Source Setup	Portuguese Verse *JOH 10:42*					
	E (*And*) CCONJ	muitos (*many*) ADJ	ali (*there*) ADV	creram (*believed*) VERB	nele (*in him*) PRON	. (.) PUNCT
English	_	ADJ	ADV	VERB	PRON	PUNCT
Spanish	CCONJ	PRON	ADV	VERB	PRON	PUNCT
French	CCONJ	_	_	VERB	PRON	PUNCT
German	CCONJ	PRON	_	VERB	_	PUNCT
Russian	CCONJ	NUM	ADV	VERB	PRON	PUNCT
Arabic	_	_	ADV	VERB	_	PUNCT
MP_{wmv}	CCONJ	PRON	ADV	VERB	PRON	PUNCT
MP_{bys}	CCONJ	ADJ	ADV	VERB	PRON	PUNCT
MP_{wbys}	CCONJ	ADJ	ADV	VERB	PRON	PUNCT

Table 5.1: Examples of single-source and multi-source projection for Finnish (upper part) and Portuguese (lower part). The alignment models are trained on the Bible.

None of the single-source assignments is fully correct. On the other hand, the weighted maximum-voting setup MP_{wmv} assigns PRON to the word *monet* in Finnish and the corresponding word *muitos* in Portuguese, where it follows the projection from Spanish and German, while the correct assignment is the one projected from English, ADJ. However, the Bayesian-inference setup MP_{bys} is able to select the ADJ assignment instead as it learns to trust English more for the projection of Adjectives, which in turn results in the overall correct POS assignment of the verse. On the other hand, the hybrid weighted Bayesian-inference setup MP_{wbys} is biased towards the selection of MP_{wmv} in the case of Finnish (PRON), while it follows MP_{bys} and produces the correct assignment

(ADJ) in the case of Portuguese, where English is a a more reliable source for adjectives.

5.2.2 Multi-Source Decoding

Since we develop multiple POS models that correspond to different source languages, we can exploit a classifier ensemble that votes among the outputs of the different models on the token level. The main difference between this approach and multi-source projection (Section 5.2.1) is that the voting takes place as part of the decoding process after applying the models on some given text in the target language, as opposed to voting among the projected tags prior to training the models. The process is illustrated in Figure 5.3.

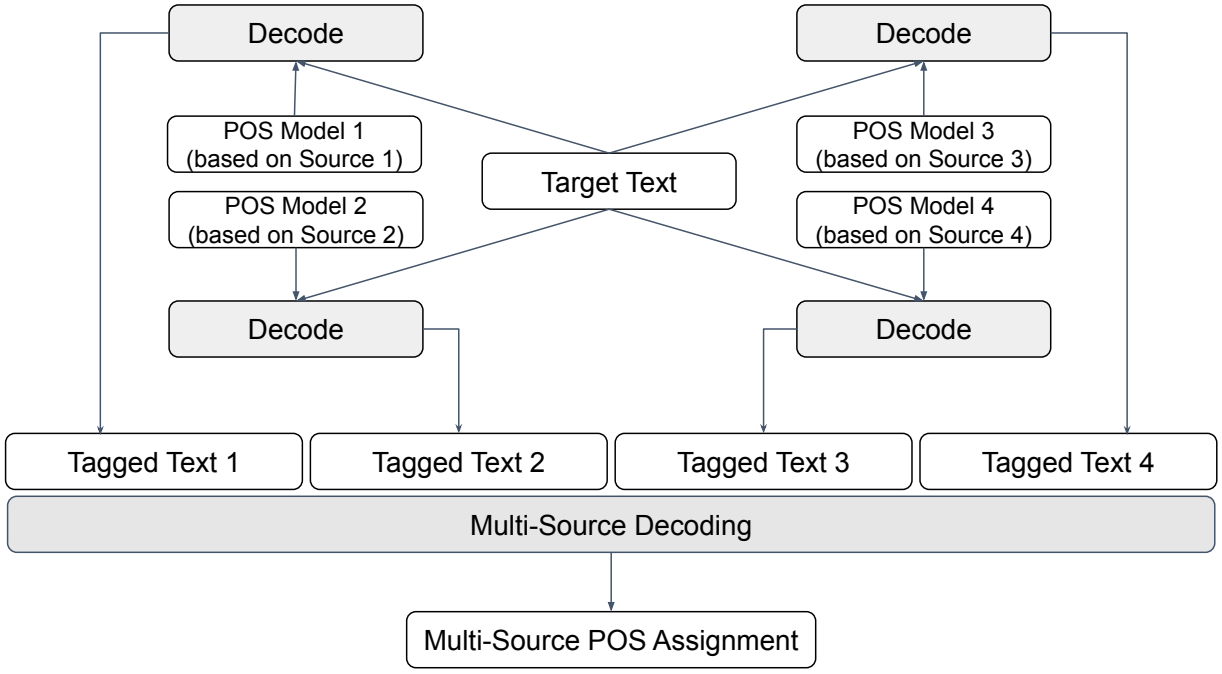


Figure 5.3: The pipeline of multi-source decoding (assuming four source languages)

We again develop three voting mechanisms for the selection of the final POS assignment, similar to the mechanisms described in Section 5.2.2: 1) weighted maximum voting (Section 5.2.2.1); 2) Bayesian inference (Section 5.2.2.2); and 3) Weighted Bayesian inference (Section 5.2.2.3). We illustrate the three mechanisms below.

5.2.2.1 Weighted Maximum Voting

We combine the tagging outputs of multiple POS models through weighted maximum voting that is similar to the MP_{wmv} setup, described in Section 5.2.1.1. Let the POS assignment of a specific {token, model} pair be y_{ij}, where $i \in [1, N]$ indexes a token, and $j \in [1, H]$ indexes a model, and let the weight of a POS assignment y_{ij} be w_{ij}. The final assignment of token i, $\hat{y}_i$, is then defined as:

$$\hat{y}_i = \arg\max_y \sum_{j,\, y_{ij}=y} w_{ij}$$

We next define the weight using two different techniques:

- **Alignment-Based Similarity**: We define the weight of a POS assignment y_{ij} as a constant value per model using a softmax function whose input vector is the average sentence-level alignment probabilities when aligning the source languages (that correspond to the POS models) to the target language prior to projecting the annotations and training the models. This weight measures the alignment-based similarity between the source language that is used for the cross-lingual learning of the underlying model and the target language on which the model is applied. In this case, we denote the voting setup by MD_{wmv_a}.

- **Decoding Probability**: We define the weight of a POS assignment y_{ij} as the softmax probability when decoding the POS tags in the final output layer of the neural models. In this case, we denote the voting setup by MD_{wmv_d}.

5.2.2.2 Bayesian Inference

We apply the Bayesian-inference setup MP_{bys}, described in Section 5.2.1.2. The main difference is that instead of measuring the reliability of each source language for the assignment of each POS tag before training the POS model, we measure the reliability of each single-source model for the assignment of each POS tag for a given text at the decoding time. We denote this setup by MD_{bys}.

5.2.2.3 Weighted Bayesian Inference

Similar to the weighted Bayesian-Inference setup MP_{bys}, described in Section 5.2.1.2, we integrate weighted maximum voting within the MD_{bys} setup at both the initialization and inference steps. We implement two setups that use the two types of weights described in Section 5.2.2.1: 1) MD_{wbys_a}, in which we define the weight as the alignment-based similarity; and 2) $\text{MD}_{\text{wbys\_d}}$, in which we define the weight as the decoding probability.

5.2.2.4 Example of Multi-Source Decoding

Table 5.2 shows a decoding example of a Basque sentence using both single-source models and multi-source decoding. The Bayesian-inference setups are able to yield the correct POS assignment of the sentence, where they trust the PRON and AUX assignments of the French and German models and thus tag the words *Zuk* and *duzu* as PRON and AUX, respectively, despite the weighted majority voting of VERB for both words.

Source Language / Multi-Source Setup	Basque Sentence				
	Zuk (*you*) PRON	*hartzen* (*take*) VERB	*duzu* (*do*) AUX	*tratamendua* (*treatment*) NOUN	*?* (*?*) PUNCT
English	PROPN	VERB	PRON	NOUN	PUNCT
Spanish	VERB	VERB	VERB	NOUN	PUNCT
French	PRON	VERB	AUX	NOUN	PUNCT
German	PRON	VERB	AUX	NOUN	PUNCT
Russian	VERB	VERB	VERB	NOUN	PUNCT
Arabic	VERB	VERB	VERB	NOUN	PUNCT
MD_{wmv_a}	VERB	VERB	VERB	NOUN	PUNCT
MD_{wmv_d}	VERB	VERB	VERB	NOUN	PUNCT
MD_{wbys}	PRON	VERB	AUX	NOUN	PUNCT
MD_{wbys_a}	PRON	VERB	AUX	NOUN	PUNCT
MD_{wbys_d}	PRON	VERB	AUX	NOUN	PUNCT

Table 5.2: A decoding example of a Basque sentence using single-source models and multi-source decoding

5.3 Languages and Data

We use the same languages and datasets as in our work on single-source cross-lingual POS tagging in Section 4.3. One main advantage is that multi-source projection increases the number of training instances. This is because multi-source projection produces null assignments that are only the overlapping ones across those of the individual sources. This leads to more dense sentences of higher scores, where the score of a sentence is defined as the harmonic mean of its density and alignment confidence (Section 4.2.1). As a result, the number of qualifying training instances increases.

Target Language	Average No. of Training Instances		Relative Increase %
	Single-Source	Multi-Source	
Afrikaans	23,784	30,877	29.8
Amharic	10,045	26,561	164.4
Basque	7,225	7,944	9.9
Bulgarian	21,600	30,407	40.8
Finnish	23,998	30,922	28.8
Georgian	7,794	7,955	2.1
Hindi	16,105	30,915	92.0
Indonesian	9,570	28,932	202.3
Kazakh	4,330	5,870	35.6
Lithuanian	25,653	31,097	21.2
Persian	17,517	30,869	76.2
Portuguese	26,751	31,076	16.2
Telugu	10,144	30,027	196.0
Turkish	16,029	30,122	87.9
Average	15,753	25,255	60.3

Table 5.3: The average number of training instances per target language, across the source languages, in the single-source setups and the multi-source projection setups when using the Bible as the source of parallel data

Table 5.3 reports the average number of training instances per target language, across the source languages, in the single-source setups and the multi-source projection setups. On average, multi-source projection results in a relative increase of 60.3% in the number of training instances, where Indonesian, Telugu and Amharic witness the highest relative increases of 202.3%, 196.0%

and 164.4%, respectively, while Georgian experiences the lowest relative increase of only 2.1%.

As the number of training instances increases, more words are seen in the training phase, and thus the percentage of out-of-vocabulary words (OOVs) decreases, which in turn improves the overall performance of the POS model.

Table 5.4 reports the average percentage of OOVs per target language, across the source languages, in the single-source setups and the multi-source projection setups. On average, multi-source projection results in a relative decrease of 7.0% in the percentage of OOVs, where Amharic, Turkish and Indonesian witness the highest relative decreases of 14.9%, 13.8% and 12.2%, respectively, while Basque experiences the lowest relative decrease of only 0.2%.

Target Language	Average Percentage of OOVs %		Relative Decrease %
	Single-Source	Multi-Source	
Afrikaans	23.3	21.6	7.1
Amharic	56.8	48.4	14.9
Basque	64.1	64.0	0.2
Bulgarian	31.6	29.4	6.9
Finnish	42.7	41.2	3.5
Georgian	38.8	38.6	0.3
Hindi	32.9	29.9	9.2
Indonesian	32.6	28.6	12.2
Kazakh	39.6	37.5	5.3
Lithuanian	36.3	34.8	4.2
Persian	33.8	31.0	8.1
Portuguese	31.0	30.1	3.2
Telugu	43.6	39.8	8.6
Turkish	34.1	29.4	13.8
Average	38.7	36.0	7.0

Table 5.4: The average percentage of out-of-vocabulary words (OOVs) per target language, across the source languages, in the single-source setups and the multi-source projection setups when using the Bible as the source of parallel data

5.4 Evaluation and Analysis

We use the same experimental settings we apply in our work on single-source cross-lingual POS tagging in Section 4.4.1, where we run the training processes for each experimental target-setup

pair for three times and report the average POS accuracy over the three runs.

5.4.1 Overall System Performance

Table 5.5 reports the accuracy of our POS taggers in the multi-source setups, compared to the best single-source setup (from Table 4.4), the average performance per multi-source setup and the average multi-source performance per target language. The last column reports the upper-bound supervised performance using *Stanza* (from Table 4.4).

Target Language	Single Source (Best)	Multi-Source Setup									Supervised (Upper Bound)
		MP_{wmv}	MP_{bys}	MP_{wbys}	MD_{wmv_a}	MD_{wmv_d}	MD_{bys}	MD_{wbys_a}	MD_{wbys_d}	Ave.	
Afrikaans	86.9	**89.1**	87.1	87.3	83.3	86.3	86.1	86.1	86.1	86.5	97.9
Amharic	75.3	**79.7**	78.4	77.4	77.6	77.3	75.8	76.5	75.9	77.3	NA
Basque	67.3	67.1	66.1	66.0	66.4	66.7	**68.6**	68.4	68.4	67.2	96.2
Bulgarian	87.2	**88.1**	87.8	87.9	86.9	87.4	87.8	87.7	87.8	87.7	98.5
Finnish	82.8	83.5	**83.6**	83.0	82.1	83.0	83.1	83.1	83.1	83.1	97.1
Georgian	83.1	**84.3**	83.2	82.8	83.6	84.1	**84.3**	84.2	84.2	83.8	NA
Hindi	73.9	72.2	72.2	72.2	**74.1**	73.4	74.0	73.9	73.9	73.2	97.6
Indonesian	84.1	82.9	83.5	83.5	84.2	**84.9**	84.7	84.6	84.7	84.1	93.7
Kazakh	**73.6**	70.3	67.4	68.4	69.7	69.4	70.7	70.7	70.7	69.7	NA
Lithuanian	**83.3**	82.9	82.5	82.0	81.5	81.6	82.0	81.9	81.9	82.0	93.5
Persian	78.1	77.3	76.3	77.0	79.0	79.0	80.1	**80.2**	80.1	78.6	81.1
Portuguese	**88.7**	87.8	87.4	88.1	88.6	88.2	88.0	88.1	88.1	88.0	92.3
Telugu	**80.0**	76.4	74.4	73.8	75.4	75.9	75.6	75.8	75.9	75.4	93.8
Turkish	74.7	74.9	73.1	72.5	74.9	**75.2**	75.1	**75.2**	**75.2**	74.5	94.7
Average	79.3	**79.7**	78.8	78.7	79.1	79.5	**79.7**	**79.7**	**79.7**	79.4	94.2

Table 5.5: The POS-tagging performance (accuracy) of the multi-source setups and the best single-source setup (from Table 4.4) when using the Bible as the source of parallel data. The best results per target language and on average, across the multi-source setups, are in **bold**. Improvement in the multi-source setups that are not statistically significant for $p\text{-}value < 0.01$ are underlined. The last column reports the upper-bound supervised performance using *Stanza*.

The multi-source approaches achieve the best on-average performance, across the target languages, and yield the best tagging performance for all the target languages except Kazakh, Lithuanian, Portuguese and Telugu. When comparing the best multi-source setup to the best single-source

setup, Amharic and Afrikaans benefit the most through the application of MP_{wmv} with relative error reductions of 17.7% and 17.1%, respectively, followed by Persian, which receives a relative error reduction of 9.2% by the application of MD_{wbys_a}. In addition, Amharic is the only target language where all the multi-source setups outperform the best single-source performance.

The improvement by multi-source projection is due to the significant increase in the number of training instances (Table 5.3) and the significant decrease in the percentage of OOVs (Table 5.4), along with the improved quality of the projected tags. On the other hand, the improvement by multi-source decoding is due to combining the outputs of different models that excel at tagging different sets of tags, where the models are based on different training sets learned through different source languages.

The best on-average multi-source performance is given by four multi-source setups, namely MP_{wmv}, MD_{bys}, MD_{wbys_a} and MD_{wbys_d}, achieving an average POS accuracy of 79.7%, across the target languages. However, MP_{wmv} is the only multi-source setup that achieves the best performance for three target languages, namely Afrikaans, Amharic and Georgian. Accordingly, we consider MP_{wmv} our best performing multi-source projection setup. As for multi-source decoding, both MD_{bys} and MD_{wbys_a} yield the best performance for two target languages. However, we consider MD_{bys} a superior model due to its simplicity, and thus it becomes our best performing multi-source decoding setup. On the contrary, MP_{wbys} and MD_{wbys_d} do not result in the best performance for any target language.

In multi-source projection, weighted maximum voting outperforms Bayesian inference except in the cases of Finnish, Indonesian and Portuguese. On the contrary, in multi-source decoding, Bayesian inference outperforms weighted maximum voting except in the cases of Afrikaans, Amharic, Indonesian and Portuguese. On another hand, the use of the decoding probability outperforms the use of the alignment-based similarity in the weighted maximum-voting setups, while the two weights result in the same on-average performance when coupled with Bayesian inference.

Finally, all the improvements due to the application of the multi-source setups, compared to the

best single-source performance, are statistically significant for $p\text{-}value < 0.01$ except in the few underlined cases in Table 4.5.

5.4.2 Performance on Open-Class Tags

Tables 5.6 and 5.7 report the precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source projection setup MP_{wmv} and the best multi-source decoding setup MD_{bys}, respectively.

Target Language	Noun			Verb			Adjective		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Afrikaans	**89.6**	91.7	90.7	77.9	92.0	84.4	85.5	**79.5**	**82.4**
Amharic	74.1	83.7	78.6	81.8	81.9	81.8	48.8	37.6	42.5
Basque	62.1	78.7	69.4	51.9	76.0	61.7	36.7	17.7	23.8
Bulgarian	88.4	**96.4**	**92.2**	86.7	89.8	88.2	85.1	58.7	69.4
Finnish	80.4	90.3	85.0	75.3	84.4	79.6	75.3	54.3	63.1
Georgian	79.1	90.6	84.5	73.9	**95.5**	83.3	**87.7**	62.8	73.1
Hindi	66.7	85.1	74.8	53.1	76.9	62.8	55.4	51.5	53.4
Indonesian	71.8	90.6	80.1	85.6	82.8	84.2	64.0	51.5	57.0
Kazakh	69.6	75.5	72.5	49.4	89.5	63.6	67.4	8.2	14.6
Lithuanian	84.4	93.8	88.9	84.8	87.5	86.1	61.7	53.7	57.4
Persian	87.6	83.3	85.4	38.5	66.5	48.8	82.9	38.4	52.5
Portuguese	89.3	94.9	92.0	**87.0**	91.0	**89.0**	78.0	74.9	76.4
Telugu	77.0	57.5	65.8	68.1	93.8	78.9	22.9	26.7	24.4
Turkish	76.5	78.8	77.6	74.6	87.5	80.6	77.7	33.7	47.0
Average	78.3	85.1	81.2	70.6	85.4	76.6	66.4	46.4	52.7

Table 5.6: The precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source projection setup when using the Bible as the source of parallel data. The best result per POS tag and evaluation metric is in **bold**.

The average F1-scores for nouns, verbs and adjectives in the MP_{wmv} setup are higher than those in the single-source setups by absolute 2.6%, 2.3% and 7.9%, respectively, while the average F1-scores for nouns, verbs and adjectives in the MD_{bys} setup are higher than those in the single-source setups by absolute 3.0%, 3.2% and 6.3%. However, the multi-source and single-source setups yield similar patterns. Two exceptions are that in the MD_{bys} setup, adjectives have a higher F1-score than verbs in Afrikaans, while verbs have a higher F1-score than nouns in Georgian.

Target Language	Noun			Verb			Adjective		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Afrikaans	**91.3**	91.5	91.4	72.9	91.3	81.1	**85.7**	**80.9**	**83.2**
Amharic	70.6	78.6	74.4	85.1	72.4	78.2	34.5	25.6	29.4
Basque	63.0	77.7	69.6	54.2	75.3	63.0	36.2	18.3	24.3
Bulgarian	90.9	**96.2**	**93.5**	86.0	88.8	87.4	85.6	62.6	72.3
Finnish	82.3	87.8	85.0	75.5	84.4	79.7	68.4	58.0	62.7
Georgian	79.2	88.0	83.3	75.0	**97.6**	84.8	**85.7**	64.1	73.3
Hindi	71.0	84.9	77.3	57.3	76.8	65.6	58.5	53.4	55.8
Indonesian	76.7	89.4	82.6	**88.7**	85.8	87.2	62.3	53.2	57.3
Kazakh	68.1	79.7	73.4	51.6	88.8	65.3	70.4	6.2	11.4
Lithuanian	85.7	92.4	88.9	86.6	83.9	85.2	56.7	62.8	59.6
Persian	89.1	83.5	86.2	42.9	73.7	54.2	83.8	47.5	60.7
Portuguese	90.7	93.8	92.2	87.2	91.2	**89.2**	75.4	76.5	75.9
Telugu	73.4	62.4	67.4	72.8	94.8	82.4	0.0	0.0	0.0
Turkish	77.5	77.8	77.6	78.0	86.9	82.2	79.3	34.0	47.6
Average	79.2	84.5	81.6	72.4	85.1	77.5	63.0	45.9	51.0

Table 5.7: The precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source decoding setup when using the Bible as the source of parallel data. The best result per POS tag and evaluation metric is in **bold**.

5.4.3 Ablation Setups

We examine the No_XLM and No_Mono ablation setups, described in Section 5.4.3, when applying the best multi-source projection setup MP_{wmv} and the best multi-source decoding setup MD_{bys}. We report the drop in POS accuracy per target language for MP_{wmv} and MD_{bys} in Figures 5.4 and 5.5, respectively.

In the No_XLM setup, the average POS accuracy across all the target languages decreases by 1.8% and 1.4% when applying multi-source projection and multi-source decoding, respectively, as opposed to an average drop of 2.2% when relying on a single source language. Similarly, in the No_Mono setup, the average POS accuracy across all the target languages decreases by 4.4% when applying either multi-source projection or multi-source decoding, as opposed to an average drop of 5.4% in the single-source setups. This means that the multi-source approaches compensate for some of the performance drop due to the lack of adequate computational resources and/or monolingual

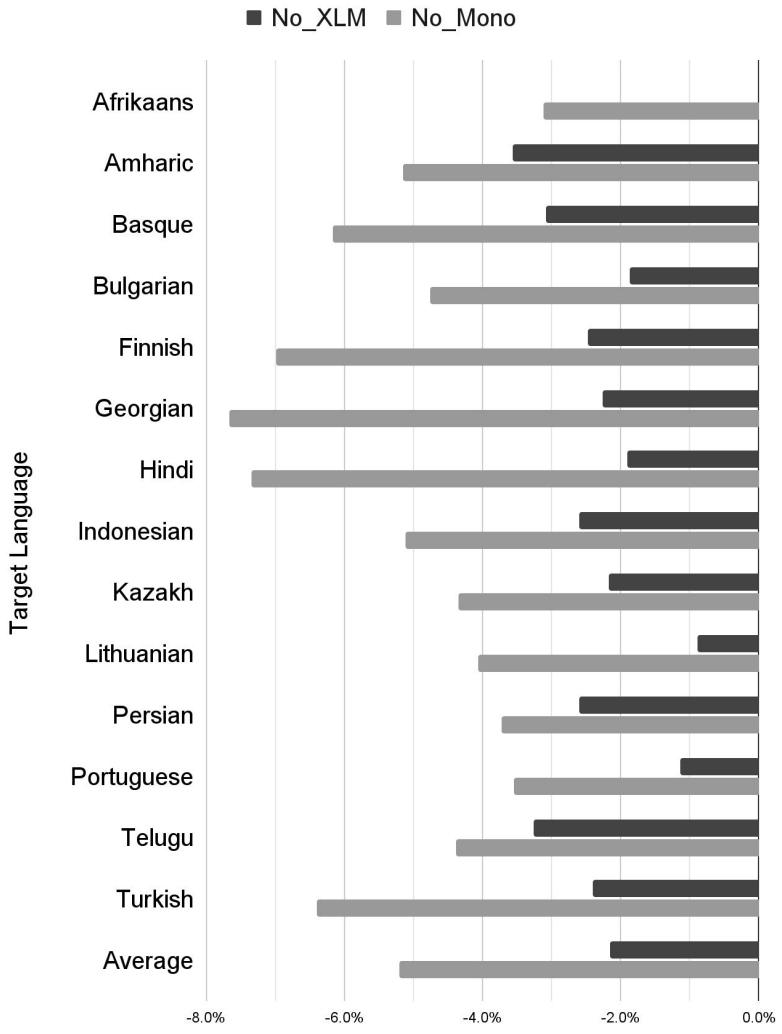


Figure 5.4: The drop in POS accuracy per target language in the best multi-source projection setup when applying the No_XLM (dark gray) and No_Mono (light gray) ablation setups

data. For example, with the elimination of the *XLM-R* model, Amharic experiences a performance drop of only 1.6% in the MP_{wmv} setup, as opposed to 3.6% when relying on a single source language. Another example is Indonesian with the ablation of the monolingual models, where it witnesses a performance drop of only 2.1% in the MD_{bys} setup, as opposed to 5.1% in the single-source setups.

One interesting phenomena is that Hindi, Indonesian and Persian experience performance increases of 1.4%, 0.7% and 0.3%, respectively, when coupling the No_XLM and MD_{bys} setups

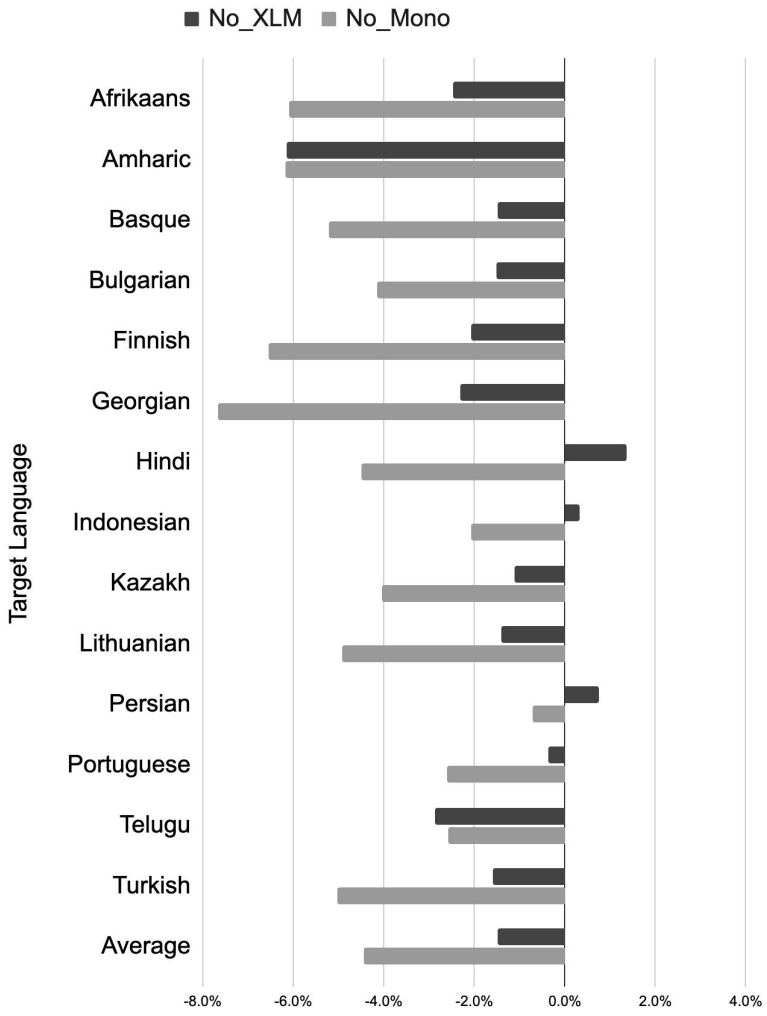


Figure 5.5: The drop in POS accuracy per target language in the best multi-source decoding setup when applying the No_XLM (dark gray) and No_Mono (light gray) ablation setups

as compared to the average performance of the regular single-source setups. One explanation is that the elimination of the multilingual *XLM-R* model helps Bayesian-inference compute confusion matrices that better learn which single-source models to trust for specific tags.

5.4.4 Comparison to State-of-the-Art

Next, we show that our multi-source approaches outperform the state-of-the-art unsupervised and semi-supervised cross-lingual POS taggers that rely on a large number of source languages to learn from.

We first compare our best multi-source projection setup MP_{wmv} and our best multi-source decoding setup MD_{bys} to the unsupervised multi-source system by Agić et al. (2016), denoted by *AGIC*. *AGIC* is a multi-source annotation-projection system that is the basis of our MP_{wmv} setup. It utilizes the Watchtower Corpus (WTC) for alignment and projection, which is a better source of parallel data than the Bible as it includes texts that are closer to contemporary language and more reliable alignments. The emission and transition probabilities are calculated based on the projected tags and are used to train a TnT POS tagger (Brants, 2000). In addition to the source of parallel data and the architecture of the POS model, a main difference between our system and *AGIC* is that *AGIC* relies on 23 source languages to learn from, as opposed to the use of only six source languages in our multi-source approaches.

We evaluate the performance of our system versus *AGIC* on the shared target languages, namely Bulgarian, Finnish, Hindi, Indonesian, Persian and Portuguese, and we report the results on the development sets of UD-v1.2 (as in the evaluation by Agić et al. (2016)) in Table 5.8. Despite the use of fewer source languages and a less suitable source of parallel data, our approach outperforms *AGIC* on all the target languages with average relative error reductions of 51.5% and 52.0% in the MP_{wmv} and MD_{bys} setups, respectively.

Target	Source	*AGIC*	MP_{wmv}	MD_{bys}
Bulgarian	Multi-source	70.0	**85.9**	84.4
Finnish	Multi-source	69.6	**80.9**	80.6
Hindi	Multi-source	50.5	71.2	**73.0**
Indonesian	Multi-source	75.5	83.4	**84.7**
Persian	Multi-source	33.7	75.0	**77.7**
Portuguese	Multi-source	84.2	**88.4**	87.2

Table 5.8: Comparison to *AGIC*, an unsupervised multi-source system for cross-lingual POS tagging, in terms of POS accuracy. The best result per target language is in **bold**.

Next, we compare our approach to the state-of-the-art multi-source semi-supervised system by Plank and Agić (2018), denoted by *DsDs*. *DsDs* follows the annotation-projection approach by Agić et al. (2016), where it replaces the TnT tagger by a BiLSTM one. The main difference between our system and *DsDs* is that *DsDs* utilizes the Polyglot embeddings (Al-Rfou' et al., 2013) and lexical information from the Wiktionary in order to control the projection process in a semi-supervised fashion.

We evaluate our system versus *DsDs* on the shared target languages, namely Basque, Bulgarian, Finnish, Hindi, Persian, and Portuguese, using the test set of UD-v2.1 for Basque and the development sets of UD-v2.1 for the other languages (as in the evaluation by Plank and Agić (2018)), and we report the performance based on the 12 universal POS tags of Petrov et al. (2012), namely ADJ (adjective), ADP (adposition), ADV (adverb), CONJ (conjunction), DET (determiner), NOUN (noun), NUM (number), PART (particle), PRON (pronoun), PUNCT (punctuation), VERB (verb) and X (a placeholder for all other tags) in Table 5.9. Despite the use of fewer source languages, a less suitable source of parallel data and a fully unsupervised approach that does not make use of external language-dependent resources, our approach outperforms *DsDs* on all the target languages except Bulgarian and Portuguese with average relative error reductions of 25.4% and 24.8% in the MP_{wmv} and MD_{bys} setups, respectively.

Target	Source	*DsDs*	MP_{wmv}	MD_{bys}
Basque	Multi-source	62.7	75.8	**76.9**
Bulgarian	Multi-source	**89.7**	89.4	89.0
Finnish	Multi-source	82.4	**85.8**	85.3
Hindi	Multi-source	66.2	82.8	**83.7**
Persian	Multi-source	43.8	78.8	**81.8**
Portuguese	Multi-source	**92.2**	91.4	90.7

Table 5.9: Comparison to *DsDs*, a multi-source semi-supervised system for cross-lingual POS tagging, in terms of POS accuracy. The best result per target language is in **bold**.

5.5 Annotation Projection vs. Supervised Learning

The comparison of the performance of the best multi-source setup of each target language to the corresponding upper-bound supervised performance by *Stanza* (from Table 5.5) shows that in four target languages, namely Afrikaans, Indonesian, Persian and Portuguese, the unsupervised multi-source taggers successfully predict at least as many as 90.0% of the correct decisions made by their corresponding supervised ones, where the percentage reaches 98.8% in the case of Persian.

Language	Annotation Size	POS Accuracy %
Afrikaans	5,700	89.1
Basque	1,300	68.6
Bulgarian	2,500	88.1
Finnish	5,800	83.6
Hindi	1,900	74.1
Indonesian	2,900	84.9
Lithuanian	6,600	83.3
Persian	2,500	80.2
Portuguese	6,900	88.7
Telugu	1,200	80.0
Turkish	2,600	75.2
Average	3,627	81.4

Table 5.10: The training size (in words) of the supervised tagger that approximates the performance of the best unsupervised single-source/multi-source setup per target language

Now, we get back to the question of "How many labeled words are needed in order to build a supervised POS tagger that approximates the performance of the unsupervised one?", discussed in Section 4.5. The use of the multi-source approaches boosts the performance in 10 target languages, and thus more labeled data is needed to train comparable supervised taggers. We follow the procedure discussed in Section 4.5 to compute the sizes of the training datasets needed to build supervised POS taggers that approximate the performance of the best unsupervised cross-lingual setup, either single-source or multi-source. The results are reported in Table 5.10. As seen, it is needed to annotate 3,364 words on average in order to develop a supervised tagger that yields the unsupervised performance, ranging from 1,200 words, in Telugu, to 6,900 words, in Portuguese.

5.6 Conclusion

In this chapter, we performed unsupervised multi-source cross-lingual POS tagging via annotation projection, where we utilize parallel data involving multiple source languages. We again used the Bible as the source of parallel data for alignment and projection in order to preserve our low-resource settings. We introduced two multi-source approaches: 1) multi-source projection, where we project the POS tags from multiple source languages before training the POS model; and 2) multi-source decoding, where we combine the outputs of different single-source POS models to tag a given text in the target language.

In order to vote among different POS tag assignments, we used two main mechanisms, weighted maximum voting and Bayesian inference. The Bayesian-inference mechanism relies on constructing confusion matrices that learn which sources to rely on for specific sets of tags. We also developed hybrid setups that perform weighted Bayesian inference. In the case of multi-source projection, we use the alignment probabilities of the underlying language pair to represent the weight, while in the case of multi-source decoding, the weights can either represent the alignment probabilities or the decoding probabilities induced by the tagging models. The different settings give a total of eight multi-source setups.

We evaluated the multi-source approaches on the 14 target languages on which we evaluated our single-source setups. Our multi-source approaches give the best performance on average and in ten target languages and improve the tagging of open-class words (nouns, verbs and adjectives). The biggest performance grains are experienced in the cases of Amharic and Afrikaans, where multi-source projection yields relative error reductions of 17.7% and 17.1%, respectively.

We showed that our multi-source approaches outperform the unsupervised system by Agić et al. (2016) despite the use of fewer source languages and a less suitable source of parallel data, achieving an average relative error reduction of 51.5%. Our approaches also outperform the state-of-the-art semi-supervised system by Plank and Agić (2018) despite the use of fewer source languages, a less suitable source of parallel data and a fully unsupervised approach that does not make use of external

language-dependent resources, achieving an average relative error reduction of 24.8%.

We conducted our ablation setups in which we 1) eliminate the use of the transformer-based contextualized word embeddings; and 2) assume limited access to monolingual data, i.e., eliminate both the transformer-based contextualized word embeddings and the word-cluster embeddings. In the two ablation setups, our multi-source approaches reduce the drop in POS accuracy as compared to the single-source setups to only 1.8% and 4.4%, respectively.

We showed that the application of our multi-source setups makes the performance of our taggers relatively closer to the performance of supervised learning, where in four target languages, our taggers can predict at least as many as 90.0% of the correct decisions made by the corresponding supervised ones. Additionally, it is needed to annotate an average of 3,627 words in order to build supervised taggers that approximate the performance of the unsupervised ones.

Chapter 6

Unsupervised Stem-Based Cross-Lingual Part-of-Speech Tagging

6.1 Overview

In cross-lingual POS tagging via annotation projection, the word structure in the source and target languages impacts the quality of the word-level alignments and the projected tags, which affects the overall performance of the ultimate POS model. This becomes problematic for languages with rich word structures where affixation is common. Work on these languages suffers from sparse alignment models that often fail to align words corresponding to the same citation form in the source and the target, where there is no one-to-one correspondence between word structures across parallel texts due to rich paradigms and translation inconsistencies.

Sparse alignment hinders the ability of a system to project the tags properly and results in null assignments on the target side. These null assignments impact the POS model, either by introducing non-continuous labeled sequences or by decreasing the number of qualifying training examples. Adding to these practical issues, the concept of word as a unit of structure has long been questioned in language sciences (Marantz, 2001). We therefore hypothesize that using the stem as the core unit of abstraction would result in better POS models for low-resource morphologically complex languages.

Our contribution is fivefold:

- We present an approach for unsupervised stem-based cross-lingual POS tagging for low-resource morphologically complex languages, where we use the *stem* as the core unit of abstraction for alignment and projection. To our knowledge, this is the first work that exploits the stem in cross-lingual and/or unsupervised POS tagging. In order to adopt a fully unsupervised approach, we use our morphological segmentation framework *MorphAGram* to

derive the stems on the target side. We also use the Bible as the source of parallel data for alignment and projection in order to preserve our low-resource settings (Section 6.2).

- We examine the use of the *morphemes* as the core unit of abstraction for alignment and projection, which allows for abstracting away from how the morphemes are combined in the source and target languages (e.g., whether they are free-standing or not). We use Arabic as the source language in our morpheme-based experiments, where we again use *MorphAGram* to derive the morphemes on the target side (Section 6.2.2).

- We examine the use of linguistic priors in morphological segmentation as a strategy for achieving better POS tagging. We use Georgian as a case study (Section 6.2.4).

- We examine the use of segmentation information (affixes and stems) as learning features in our neural architecture (Section 6.2.5).

- We conduct extensive evaluation and analysis using eight morphologically complex target languages out of the 14 languages we evaluate our word-based approach on (Sections 4.3 and 5.3), namely Amharic, Basque, Finnish, Georgian, Indonesian, Kazakh, Telugu and Turkish, along with the same set of source languages. We evaluate our stem-based models in both the single-source and multi-source setups (Sections 6.3 and 6.4).

We show that the stem-based approach outperforms the word-based one in 43 language pairs out of the 48 pairs we experiment with, achieving average relative error reductions up to 21.4% in the case of Kazakh. We also show that the stem-based approach outperforms the word-based one that operates on three-times more data in about two thirds of the pairs we experiment with. In addition, we illustrate that the multi-source setups also benefit from the stem-based approach despite the relatively high word-based multi-source baseline, where the use of the stem for alignment and projection yields improvements in 57 multi-source pairs out of 64. As for the morpheme-based approach, we experiment with Arabic as the source language and report improvements for all the target languages except Amharic. Additionally, we illustrate that the use of linguistic priors in

the form of affixes compiled by an expert in the underlying language, where the segmentation is performed in a semi-supervised manner, improves the performance for Georgian, as a case study. Finally, we show that the stem-based approach noticeably improves the detection of open-class tags (nouns, verbs and adjectives) in both the single-source and multi-source setups (Section 6.4).

6.2 Methods

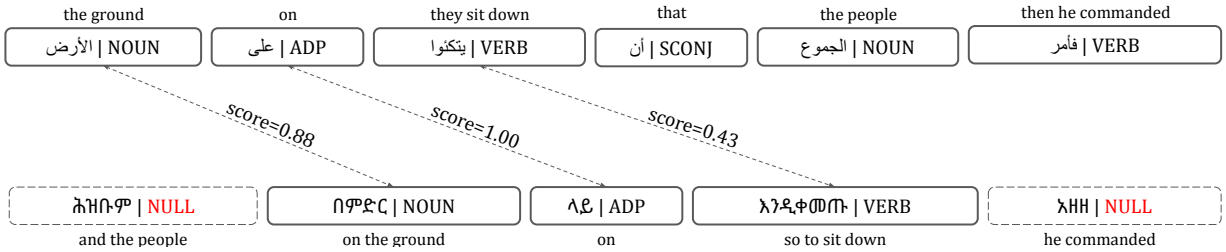
While our word-based architecture for cross-lingual POS tagging via annotation projection yields the state-of-the-art results for unsupervised POS tagging when evaluated on 14 languages of diverse typologies, the complexity of word structure in the source and target languages has a direct impact on the quality of the alignment and projection phases.

6.2.1 Challenges with Word-Based Alignment and Projection

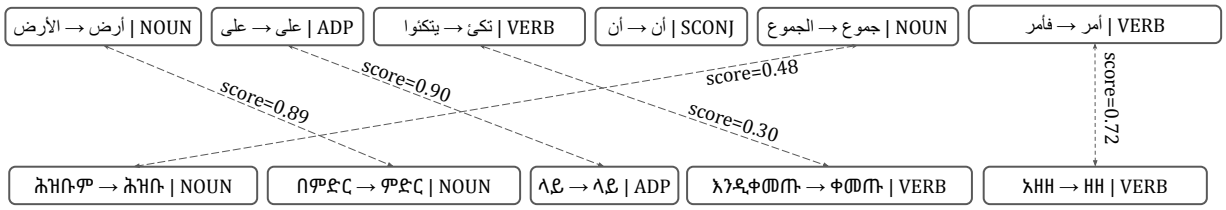
Rich word structure with excessive affixation increases the ratio of word types to word tokens, which in turn results in sparse alignment models and incomplete projections that form null assignments on the target side. Null assignments either introduce missing information for the learning of the POS model or result in scores that are too low for the underlying sentences to qualify as training instances, which negatively impacts the overall quality of the POS model.

An example is shown in Figure 6.1a, where Arabic and Amharic are the source and target languages, respectively. The example corresponds to verse *MAT 15:35* in the Bible, *"He commanded the multitude to sit down on the ground"*, where the word-alignment models are trained on the New Testament. As shown, two Arabic-Amharic word pairs are not aligned and produce null assignments, which is the result of sparse word-alignment models that are unable to align words that correspond to the same citation form properly. This happens because a single citation form might have an extensive paradigm in either language, which, along with translation inconsistencies, leads to the loss of a one-to-one correspondence between word structures across parallel texts.

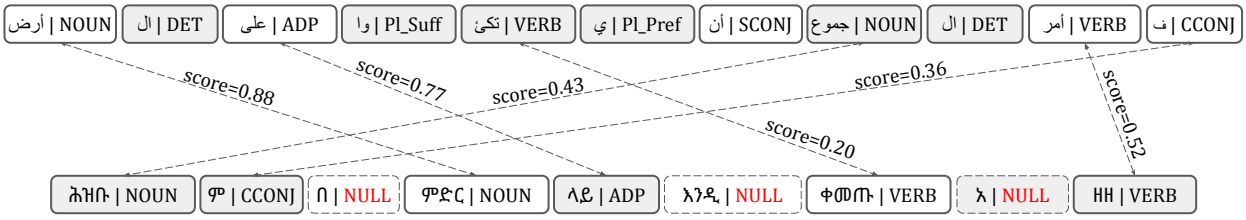
Table 6.1 shows examples of paired inflected forms that correspond to the same citation forms in Arabic and Amharic but receive different types of affixation, which in turn leads to the unaligned



(a) Word-based alignment and projection



(b) Stem-based alignment and projection



(c) Morpheme-based alignment and projection

Figure 6.1: An example of alignment and projection from Arabic onto Amharic. The alignment models are trained on the New Testament. Arabic reads right to left.

Verse	Arabic Word		Amharic Word	
MAT 15:35	الجموع	(*the people*)	ሕዝቡም	(*and the people*)
MAT 26:55	للجموع	(*to the people*)	ለሕዝቡ	(*to the people*)
LUK 9:11	فالجموع	(*and the people*	ሕዝቡም	(*and the people*)
LUK 23:4	والجموع	(*and the people*)	ለሕዝቡ	(*to the people*)
MAT 1:24	أمره	(*he commanded him*)	እንዳዘዘው	(*as he commanded*)
MAT 15:35	فأمر	(*then he commanded*)	አዘዘ	(*he commanded*)
b.LUK 5:14	أمر	(*he commanded*)	እንዳዘዘ	(*as he commanded*)
b.ACT 21:34	أمر	(*he commanded*)	አዘዘ	(*he commanded*)

Table 6.1: Paired inflected forms that correspond to the same citation form across Arabic and Amharic parallel verses in the New Testament

pairs in Figure 6.1a.

We next show that using the stem as the core unit of abstraction for alignment and projection improves POS tagging (Section 6.2.2). In addition, we examine the use of the morphemes instead of the stem as the abstraction unit when the source and target languages are morphologically complex (Section 6.2.3). Moreover, we examine the use of linguistic priors towards better segmentation and tagging models (Section 6.2.4). Finally, we exploit the segmentation output as learning features in our neural architecture (Section 6.2.5).

6.2.2 Stem-Based Alignment and Projection

Using the stem instead of the word as the core unit of abstraction is more productive; the stem is usually shared by all the members of a paradigm, which allows to minimize misalignment.

Figure 6.1b shows that stemming the Arabic and Amharic texts results in complete one-to-one alignments and projections, which in turn eliminates the null assignments resulting from the word-level approach and assigns each word on the Amharic side a valid POS assignment.

Figure 6.2 illustrates our overall pipeline of the stem-based approach. First, we conduct stemming for the source and target texts, and then we train stem-based alignment models between the two sides. Next, we assign the stems of the source side the POS tags of their corresponding words, which are then projected onto the target stems using the bidirectional stem-based alignments. We then apply the token and type constraints on the labeled stems on the target side. However, since we train the ultimate POS model on the word level, we replace each target stem by its corresponding word and assign the word the stem-based POS tag. The rest of the pipeline for sentence selection and training the POS model is the same as in the word-based architecture described in Section 4.2.

We assume that the source language is a high-resource one for which an off-the-shelf stemmer is accessible, where we use the *Snowball* Stemmer (Porter, 2001) as part of *NLTK* [1] (Bird and Loper, 2004) for the stemming of English, Spanish, French, German and Russian, while we use *MADAMIRA* for Arabic, for performance gain. On the other hand, we apply our morphological-

[1] https://www.nltk.org

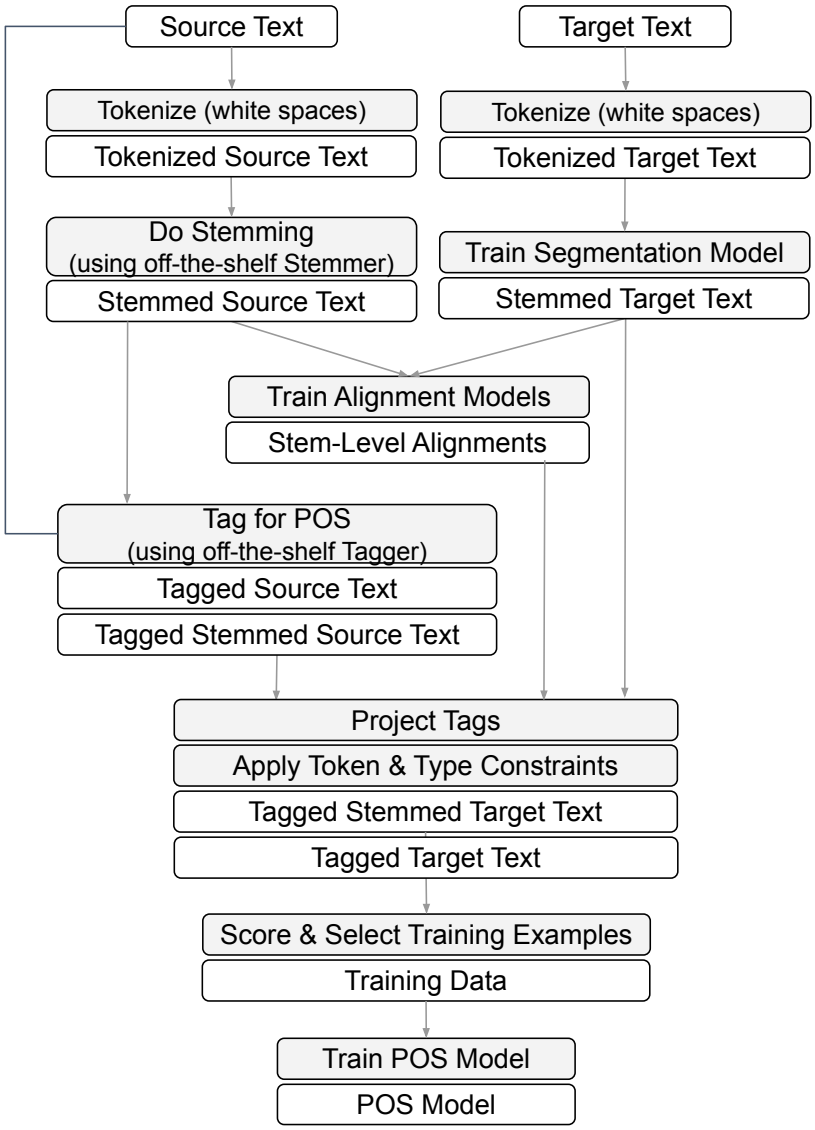


Figure 6.2: The overall pipeline of unsupervised word-based cross-lingual POS tagging via annotation projection

segmentation framework *MorphAGram* on the target side of the parallel text in a fully unsupervised manner. We run *MorphAGram* in the cascaded setting (Section 3.3) using two rounds of learning. In the first round, we train a segmentation model using the language-independent high-precision grammar *PrStSu2b+Co+SM* to obtain a list of affixes. We then seed these affixes into the best performing language-independent grammar *PrStSu+SM* for the second round of learning. As discussed in Section 3.3, both *PrStSu2b+Co+SM* and *PrStSu+SM* model the word as a sequence of

prefixes, a stem and suffixes, where the prefixes and suffixes are recursively defined in order to model multiple consecutive affixes, which is suitable to process morphologically complex languages.

6.2.3 Morpheme-Based Alignment and Projection

Next, we examine the use of morpheme-level alignment and projection, similar to the stem-based approach in Section 6.2.2. This approach abstracts away from whether the morphemes in the source and target languages are free-standing or not. We hypothesize that morpheme-based alignment and projection would help when both the source and target languages are morphologically complex, and therefore we examine this approach using Arabic as the source.

Figure 6.1c shows that conducting both alignment and projection on the morpheme level for Arabic and Amharic results in a complete POS assignment on the Amharic side as at least one morpheme from each word receives a POS tag, where every sequence of consecutive morphemes in the same color corresponds to one word.

The pipeline of the morpheme-based approach is similar to the one of the stem-based approach depicted in Figure 6.2. On the source side, each morpheme receives a separate POS tag using an off-the-shelf POS tagger, where we use *MADAMIRA* to obtain the morphemes for Arabic, while we obtain the target morphemes in inflected forms (morphs) using *MorphAGram* [2] by applying the same cascaded setting we use in the stem-based approach. We then project the POS tags from the source morphemes onto the target ones through bidirectional morpheme-level alignments that are induced by morpheme-level alignment models trained in the morpheme space. Upon applying the token and type constraints, since we train the POS model on the word level, we replace each sequence of morphemes that corresponds to one word on the target side by its corresponding word and assign that word the POS tag of the representative morpheme. We define the representative morpheme either as the morpheme whose POS tag ranks the highest among those of the other morph [3] (*RANK*) or as the stem morpheme (*STEM*). For instance, the first Amharic word in Figure 6.1c receives the

[2]While the use of morphemes is more appealing than morphs towards less sparse models, we use morphs on the target side as the extraction of morphemes is not plausible in a fully unsupervised manner using *MorphAGram*.

[3]We use the default ranking of POS tags defined at https://github.com/coastalcph/ud-conversion-tools. That is VERB, NOUN, PROPN, PRON, ADJ, NUM, ADV, INTJ, AUX, ADP, DET, PART, CONJ, SCONJ, X and PUNCT.

NOUN tag either because NOUN supersedes CCONJ (*RANK*) or because NOUN is assigned to the stem morpheme (*STEM*).

6.2.4 Stem-Based Approach with Linguistic Priors

We hypothesize that better detection of stems in the stem-based approach would yield more robust alignment and projection, which in turn results in a better POS model. Accordingly, instead of conducting morphological segmentation on the target side in a fully unsupervised manner, we incorporate linguistic priors in the form of linguist-provided affixes as detailed in Section 3.3.4. In this setup, a list of affix morphemes is compiled manually by an expert in the target language and seeded into the *PrStSu+SM* grammar prior to training the segmentation model.

6.2.5 Segmentation Information as Learning Features

Next, we examine the use of segmentation features within our neural architecture. We use the unsupervised morphological-segmentation model that is trained on the target side of the parallel text to produce stem, complex-prefix and complex-suffix features for each target word. For training, we use these features as randomly initialized embeddings that we concatenate with the existing word, affix and words-cluster embeddings to represent the input words in our BiLSTM neural architecture (Section 4.2.2), while for decoding, we apply our segmentation model on the input text to construct word representations in a similar fashion.

6.3 Languages and Data

We select eight morphologically complex target languages out of the 14 languages on which we evaluate our word-based approach (Sections 4.3 and 5.3): six morphologically complex languages that are largely agglutinative, namely Basque, Finnish, Georgian, Kazakh, Telugu, and Turkish, morphologically rich Amharic, where many morphological alterations rely on consonantal roots, and less morphologically rich Indonesian. On the other side, we use the same set of source languages. This makes a total of 48 language pairs. In addition, we experiment with the eight multi-source

setups presented in Section 5.2.

We choose to use the New Testament instead of the entire Bible as the source of parallel data for alignment and projection in the stem-based approach. This is in order to demonstrate the efficiency of stem-based alignment and projection, where the use of the stem compensates for the lack of adequate parallel data. We however use the same evaluation datasets as before.

The stem-based approach results in less-sparse alignment models, allowing for more POS projections from the source onto the target, and thus the number of null assignments decreases. This leads to more dense sentences of higher scores, where the score of a sentence is defined as the harmonic mean of its density and alignment confidence (Section 4.2.1). As a result, the number of qualifying training instances increases.

Table 6.2 reports the average number of training instances per target language, across the source languages, in the word-based and stem-based approaches. On average, the stem-based approach results in a relative increase of 16.4% in the number of training instances, where Amharic witnesses the highest relative increase of 132.6%, while Georgian experiences the lowest relative increase of only 1.9%.

Target Language	Average No. of Training Instances		Relative Increase %
	Word-Based	Stem-Based	
Amharic	2,605	6,060	132.6
Basque	7,225	7,505	3.9
Finnish	7,125	7,518	5.5
Georgian	7,794	7,942	1.9
Indonesian	5,286	5,914	11.9
Kazakh	4,330	5,268	21.7
Telugu	4,719	5,382	14.1
Turkish	6,280	7,196	14.6
Average	5,670	6,598	16.4

Table 6.2: The average number of training instances per target language, across the source languages, in the word-based and stem-based approaches when using the New Testament as the source of parallel data

As the number of training instances increases, more words are seen in the training phase, and

thus the percentage of out-of-vocabulary words (OOVs) decreases, which in turn improves the overall performance of the POS model.

Table 6.3 reports the average percentage of OOVs per target language, across the source languages, in the word-based and stem-based approaches. On average, the stem-based approach results in a relative decrease of 2.6% in the percentage of OOVs, where Amharic witnesses the highest relative decrease of 9.2%, while Basque experiences the lowest relative decrease of only 0.1%.

Target Language	Average Percentage of OOVs %		Relative Decrease %
	Word-Based	Stem-Based	
Amharic	66.6	60.5	9.2
Basque	64.1	64.1	0.1
Finnish	47.0	46.7	0.7
Georgian	35.2	34.6	1.6
Indonesian	38.8	38.6	0.3
Kazakh	39.6	38.2	3.5
Telugu	46.2	45.9	0.7
Turkish	38.2	37.2	2.7
Average	47.0	45.7	2.6

Table 6.3: The average percentage of out-of-vocabulary words (OOVs) per target language, across the source languages, in the word-based and stem-based approaches when using the New Testament as the source of parallel data

Finally, we evaluate the morpheme-based approach when projecting from Arabic as it is the most morphologically complex source language, while we examine the use of linguistic priors within the stem-based approach using Georgian as a case study.

6.4 Evaluation and Analysis

We use the same experimental settings we apply in our work on word-based cross-lingual POS tagging in Section 4.4.1, where we run the training processes for each experimental pair for three times and report the average POS accuracy over the three runs.

6.4.1 Performance of Single-Source Stem-Based Setups

Table 6.4 reports the accuracy of our POS taggers in the single-source stem-based setups, compared to the single-source word-based setups, and the average relative error reductions due to the use of the stem-based approach per target language, across the source languages, and per source language, across the target languages.

Target Language	Approach	Source for Unsupervised Learning						Ave. Relative Error Reduction %
		English	Spanish	French	German	Russian	Arabic	
Amharic	Word-based	75.9	74.9	75.5	76.4	72.1	72.6	
	Stem-based	**79.6***	**77.5**	**77.7**	**77.8**	**76.2**	**74.5**	10.5
Basque	Word-based	67.3	64.6	65.8	66.7	61.7	55.6	
	Stem-based	**69.1**	**70.4***	**70.5**	**69.6**	**65.2**	**60.8**	10.9
Finnish	Word-based	81.0	78.8	77.4	79.8	77.8	66.1	
	Stem-based	**81.9**	**80.1**	**80.9***	**82.3**	**79.0**	**70.3**	9.4
Georgian	Word-based	**82.8**	80.1	80.2	**82.5**	83.1	71.2	
	Stem-based	82.0	**80.4**	**81.0**	82.2	**83.4**	**79.0***	4.6
Indonesian	Word-based	82.3	**81.6**	**81.0**	77.1	76.8	69.8	
	Stem-based	**82.5**	81.0	80.1	**77.3**	**81.2***	**72.3**	3.5
Kazakh	Word-based	73.6	64.7	67.3	68.9	62.1	63.6	
	Stem-based	**76.4**	**74.8**	**75.5**	**73.2**	**73.6***	**70.8**	21.4
Telugu	Word-based	76.7	68.4	67.9	70.4	63.5	59.5	
	Stem-based	**78.6**	**72.7**	**72.2**	**71.9**	**69.6**	**66.8**	12.4
Turkish	Word-based	**73.9**	70.1	70.5	69.2	66.2	64.7	
	Stem-based	73.7	**73.1**	**73.0**	**71.9**	**77.6***	**71.9**	13.4
Ave. Error Reduction %		5.0	10.4	10.5	6.8	16.3	15.6	

Table 6.4: The POS-tagging performance (accuracy) of the single-source word-based and stem-based setups when using the New Testament as the source of parallel data. The best result per target-source language pair is in **bold**. The highest relative error reduction in the stem-based approach per target language is marked by *. The improvements in the stem-based setups that are not statistically significant for $p\text{-}value < 0.01$ are underlined. The last column and row report the stem-based average relative error reductions per target language and source language, respectively.

The single-source stem-based approach outperforms the single-source word-based one in 43

language pairs, where only five language pairs benefit more from word-based alignment and projection, namely {Georgian, English}, {Georgian, German}, {Indonesian, Spanish}, {Indonesian, French} and {Turkish, English}, where Georgian and Indonesian stand out in our language sample as the least complex in terms of morphology.

Comparing to the single-source word-based approach, the biggest improvements in the single-source stem-based approach are achieved in the cases of the {Turkish, Russian}, {Kazakh, Russian}, {Kazakh, Spanish} and {Georgian, Arabic} language pairs with average relative error reductions of 33.8%, 30.2%, 28.6% and 27.2%, respectively. When averaging across the source languages, Kazakh and Turkish experience the highest average relative error reductions of 21.4% and 13.4%, respectively, while the least morphologically complex Indonesian benefits from the stem-based approach the least with an average relative error reduction of only 3.5%. On the other hand, when averaging across the target languages, Russian and Arabic yield the highest average relative error reductions of 16.3% and 15.6%, respectively, which is in line with the fact that Arabic and Russian are more morphologically complex than the other source languages.

Comparing to applying the single-source word-based approach on the entire Bible (in Table 4.4), the single-source stem-based approach that relies on the New Testament, that is one fourth the amount of data, achieves better results in about two thirds of the language pairs (31 out of 48) and on average with an average absolute performance increase of 1.7% that goes up to 11.5% in the {Kazakh, Russian} language pair. This means the use of the stem as the core unit of abstraction compensates for the lack of adequate parallel data to learn from as it produces less-sparse alignment models and increases the number of projected tags and training instances. For the absolute increases in all the language pairs, see Figure 6.3.

Finally, all the improvements due to the use of the single-source stem-based approach as compared to the single-source word-based approach are statistically significant for $p\text{-}value < 0.01$ except for the {Indonesian, English}, {Indonesian, German} and {Kazakh, Spanish} language pairs. It is also noticed that for some target languages, the best source language in the word-based approach differs from the best source language in the stem-based approach. For instance, while English is

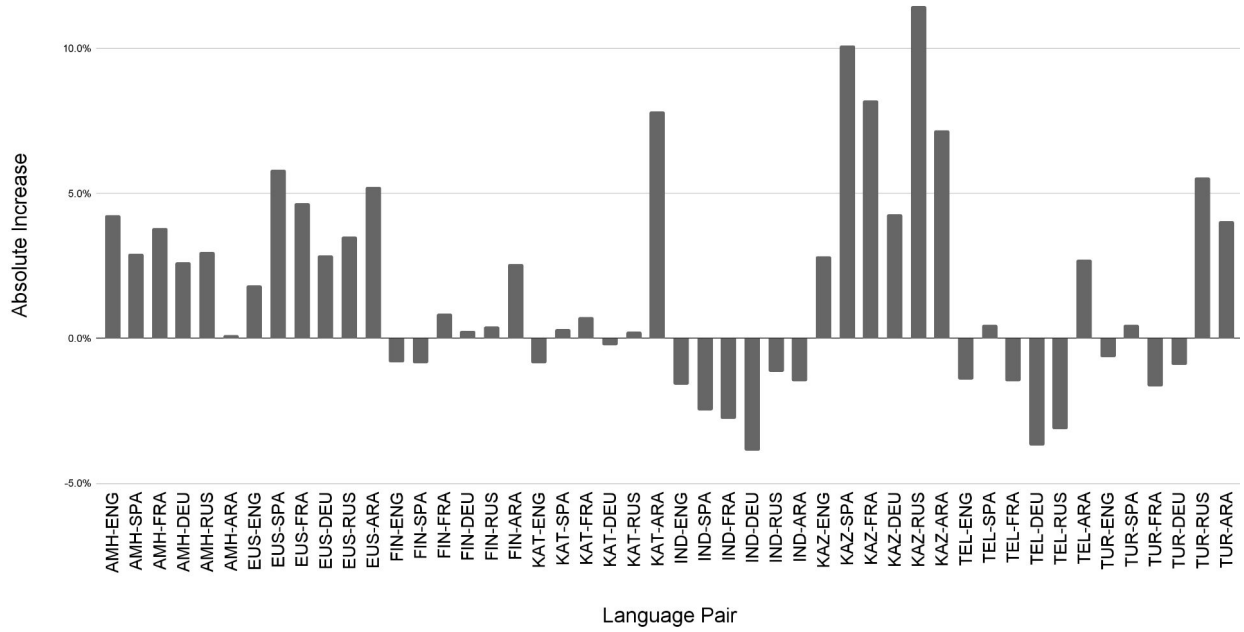


Figure 6.3: The absolute performance increases (accuracy) when applying the single-source stem-based approach using the New Testament as the source of parallel data as compared to the single-source word-based approach using the entire Bible as the source of parallel data

the best source language for Basque, Finnish and Turkish in the word-based approach, exploiting the stem gives the best performance for the three languages when projecting from Spanish/French, German and Russian, respectively.

6.4.2 Performance of Multi-source Stem-Based Setups

Table 6.5 reports the accuracy of our POS taggers in the multi-source stem-based setups, compared to the multi-source word-based setups, and the average relative error reductions due to the use of the stem-based approach per target language, across the multi-source setups, and per multi-source setup, across the target languages.

The multi-source stem-based approach outperforms the multi-source word-based one in 57 multi-source pairs out of 64. This is mainly because the least morphologically complex Indonesian does not benefit from the multi-source stem-based approach except when applying the MP_{wmv_a} setup.

Target Language	Approach	Multi-Source Setup								Ave. Relative Error Reduction %
		MP_{wmv}	MP_{bys}	MP_{wbys}	MD_{wmv_a}	MD_{wmv_d}	MD_{bys}	MD_{wbys_a}	MD_{wbys_d}	
Amharic	Word-Based	78.0	74.8	74.9	76.6	77.1	76.3	76.4	76.4	
	Stem-Based	**79.6**	**80.4**	**80.8***	**78.6**	**79.1**	**79.2**	**79.2**	**79.2**	13.3
Basque	Word-Based	67.1	66.1	66.0	66.4	66.7	68.6	68.4	68.4	
	Stem-Based	**71.4**	**71.7**	**71.7***	**71.0**	**72.0**	**71.9**	**72.3**	**71.8**	7.0
Finnish	Word-Based	81.7	82.0	81.9	81.0	81.1	81.5	81.4	81.4	
	Stem-Based	**82.9**	**82.7**	**82.5**	**82.4**	**82.7**	**83.2***	**83.0**	**83.0**	13.7
Georgian	Word-Based	84.3	83.2	82.8	83.6	84.1	84.3	84.2	84.2	
	Stem-Based	**84.7**	**84.5**	**84.2***	**84.3**	**84.3**	**84.5**	**84.4**	**84.5**	3.7
Indonesian	Word-Based	**81.7**	**81.1**	**81.2**	80.9	**82.3**	**82.2**	**82.2**	**82.3**	
	Stem-Based	81.0	81.0	80.9	**81.4***	81.9	82.0	82.0	82.0	-1.2
Kazakh	Word-Based	70.3	67.4	68.4	69.7	69.4	70.7	70.7	70.7	
	Stem-Based	**76.7**	**76.8***	**76.9**	**75.3**	**75.8**	**76.7**	**76.6**	**76.6**	22.3
Telugu	Word-Based	71.3	70.3	69.3	68.6	69.9	71.1	71.3	71.0	
	Stem-Based	**73.8**	**71.7**	**70.8**	**72.9***	**73.6**	**73.4**	**73.4**	**73.4**	8.5
Turkish	Word-Based	73.3	70.9	71.7	71.0	72.3	73.2	73.1	73.2	
	Stem-Based	**73.6**	**73.6**	**73.7**	**75.4***	**75.2**	**74.4**	**74.4**	**74.3**	7.0
Ave. Error Reduction %		7.1	11.6	11.1	10.6	9.6	8.1	8.2	8.0	

Table 6.5: The POS-tagging performance (accuracy) of the multi-source word-based and stem-based setups when using the New Testament as the source of parallel data. The best result per {target and multi-source setup} pair is in **bold**. The highest relative error reduction in the stem-based approach per target language is marked by *. The improvements in the stem-based setups that are not statistically significant for $p\text{-}value < 0.01$ are underlined. The last column and row report the stem-based average relative error reductions per target language and multi-source setup, respectively.

Comparing to the multi-source word-based approach, the biggest improvements in the multi-source stem-based approach are achieved in the cases of Kazakh and Amharic in the MP_{bys} and MP_{wbys} setups with relative error reductions of 28.9% and 27.0%, respectively, for Kazakh and 22.3% and 23.5%, respectively, for Amharic. When averaging across the multi-source setups, Kazakh and Finnish experience the highest average relative error reductions of 22.3% and 13.7%, respectively, while the performance for the least morphologically complex Indonesian witnesses an average relative drop of 1.2%. On the other hand, when averaging across the target languages, the

MP$_{bys}$and *MP*$_{wbys}$ setups yield the highest average relative error reductions of 11.6% and 11.1%, respectively, which means that the stem-based approach is more efficient for multi-source projection than multi-source decoding.

Comparing to applying the multi-source word-based approach on the entire Bible (in Table 5.5), the multi-source stem-based approach that relies on the New Testament, that is one fourth the amount of data, achieves better results in about three fifths of the pairs (37 out of 64) and on average with an average absolute performance increase of 1.1% that goes up to 9.4% in the {Kazakh, *MP*$_{bys}$} pair, which is in line with the patterns seen in the single-source setup.

Finally, all the improvements due to the use of the multi-source stem-based approach as compared to the multi-source word-based approach are statistically significant for $p\text{-}value < 0.01$ except for the {Georgian, *MD*$_{bys}$} and {Telugu, *MP*$_{bys}$} pairs.

6.4.3 Performance of Morpheme-Based Setups

Next, we evaluate the morpheme-based approach when projecting from Arabic, our source language of the richest morphology, using the *RANK* and *STEM* mechanisms for the selection of the representative morphemes. We compare the results to those of the word-based and stem-based approaches and report them in Table 6.6.

The morpheme-based approach results in more dense training instances as both alignment and projection are performed in a more fine-grained level than those in the word-based and stem-based approaches. It therefore yields the best performance for all the target languages when projecting from Arabic except Amharic, where Telugu benefits the most with relative error reductions of 23.9% and 15.3% over the stem-based approach using the *RANK* and *STEM* mechanisms, respectively. The difference in the performance of the *RANK* and *STEM* mechanisms is only statistically significant for $p\text{-}value < 0.01$ in Amharic and Basque, where the *STEM* mechanism yields better performance, and in Telugu, where the *RANK* mechanism gives a better result[4]. However, all the improvements in the morpheme-based approach as opposed to the stem-based one are statistically significant.

[4]The quality of morphological segmentation affects the detection of stems and thus affects the quality of the *STEM* mechanism.

Target Language	Approach			
	Word-Based	Stem-Based	Morpheme-Based (*RANK*)	Morpheme-Based (*STEM*)
Amharic	72.6	**74.5**	72.5	73.6
Basque	55.6	60.8	61.9	**62.2**
Finnish	66.1	70.3	73.8	**74.2**
Georgian	71.2	79.0	**80.5**	80.0
Indonesian	69.8	72.3	75.5	**75.6**
Kazakh	63.6	70.8	71.8	**71.9**
Telugu	59.5	66.8	**74.7**	71.8
Turkish	64.7	71.9	73.2	**73.4**

Table 6.6: The POS-tagging performance (accuracy) of the word-based, stem-based and morpheme-based approaches when projecting from Arabic using the New Testament as the source of parallel data. The best result per target language is in **bold**. The improvements in the morpheme-based setups that are not statistically significant for $p\text{-}value < 0.01$ are underlined.

6.4.4 Performance of Using Linguistic Priors

Next, we evaluate the use of linguistic priors in the stem-based approach, where we compile a list of linguist-provided affixes of Georgian, as a case study, and seed them into the *PrStSu+SM* grammar. We use the same list of affixes we experiment with in Section 3.3.2 and report the results in Tables 6.7 and 6.8 for the single-source and multi-source setups, respectively.

Target Language	Approach	Source for Unsupervised Learning					
		English	Spanish	French	German	Russian	Arabic
Georgian	Word-based	82.8	80.1	80.2	**82.5**	83.1	71.2
	Stem-based	82.0	80.4	81.0	82.2	83.4	**79.0**
	LP Stem-based	**82.9**	**80.8**	**82.2**	82.4	**83.9**	77.4

Table 6.7: The POS-tagging performance (accuracy) of the single-source word-based and stem-based (with and without linguistic priors (LP)) setups when using the New Testament as the source of parallel data. The best result per source language is in **bold**. The improvements in the LP stem-based setups that are not statistically significant for $p\text{-}value < 0.01$ as compared to the regular stem-based setups are underlined.

The use of linguistic-priors improves the single-source stem-based approach when projecting from all the source languages except Arabic. The linguistic priors result in an average relative error reduction of 1.7% that goes up to 6.3% when projecting from French.

Target Language	Approach	Multi-Source Setup							
		MP_{wmv}	MP_{bys}	MP_{wbys}	MD_{wmv_a}	MD_{wmv_d}	MD_{bys}	MD_{wbys_a}	MD_{wbys_d}
Georgian	Word-Based	84.3	83.2	82.8	83.6	84.1	84.3	84.2	84.2
	Stem-Based	84.7	84.5	84.2	84.3	84.3	84.5	84.4	84.5
	LP Stem-based	**85.1**	**84.9**	**84.8**	**85.3**	**85.3**	**85.4**	**85.4**	**85.4**

Table 6.8: The POS-tagging performance (accuracy) of the multi-source word-based and stem-based (with and without linguistic priors (LP)) setups when using the New Testament as the source of parallel data. The best result per multi-source setup is in **bold**. The improvements in the LP stem-based setups that are not statistically significant for $p\text{-}value < 0.01$ as compared to the regular stem-based setups are underlined.

The improvements due to the use of linguistic priors in the single-source stem-based approach are statistically significant for $p\text{-}value < 0.01$ when only projecting from English and French. This suggests that the robust approach for selecting high-quality alignments, projections and training instances limits the effect of developing a segmentation model of a relatively better quality.

The lack of improvement in the case of Arabic can be explained by over-segmentation that produces an incorrect POS tag for the conjunction და (*and*). The characters და also correspond to a verbal prefix that is manually seeded as a prior. This seeding causes erroneous projections labeling და as a verb or an adverb when projecting from Arabic.

In the multi-source setups, the use of linguistic priors consistently improves the performance, achieving an average relative error reduction of 4.6% that goes up to 6.3% in the MD_{wmv_d} setup, where the improvements in multi-source decoding are consistently higher than those in multi-source projection. Finally, all the improvements are statistically significant for $p\text{-}value < 0.01$ except in the MP_{wmv} setup.

6.4.5 Performance of Using Segmentation Features

Next, we evaluate the use of the segmentation output (stems and affixes) as learning features in our neural architecture. Overall, the majority of the improvements due to the use of these features are not statistically significant since such features are surpassed by the prefix and suffix n-gram

character-based features, where n is in $\{1, 2, 3, 4\}$ [5]. We report below the statistically significant improvements for $p\text{-}value < 0.01$. For the complete results in the single-source and multi-source setups, see Tables 2.5 and 2.6, respectively, in Appendix B.

In the single-source stem-based approach, the use of stems as features only yields statistically significant improvements for four target-source language pairs, namely {Amharic, English}, {Basque, Russian}, {Finnish, Arabic} and {Georgian, French}, where the {Amharic, English} language pair experiences the highest absolute performance increase of only 0.6%. On the other hand, coupling stems with complex prefixes and suffixes as features only results in statistically significant improvements for five target-source language pairs, namely {Basque, Arabic}, {Basque, Russian}, {Finnish, Arabic}, {Finnish, French} and {Kazakh, English}, where the {Basque, Arabic} language pair experiences the highest absolute performance increase of 1.2%

In the multi-source stem-based approach, the use of segmentation features is more beneficial than in the single-source stem-based approach, where the use of stems as features yields statistically significant improvements for seven experimental pairs, namely {Amahric, MP_{bys}}, {Amahric, MD_{wbys_d}}, {Basque, MP_{bys}}, {Basque, MD_{bys}}, {Basque, MD_{wbys_d}}, {Finnish, MD_{wmv_a}} and {Kazakh, MP_{bys}}, where projecting onto Basque in the MD_{wbys_d} setup experiences the best absolute performance gain of 0.9%. On the other hand, coupling stems with complex prefixes and suffixes as features yields statistically significant improvements for ten experimental pairs, namely {Basque, MP_{wbys}}, {Finnish, MD_{wmv_a}}, {Finnish, MD_{wmv_d}}, {Kazakh, MP_{bys}}, {Kazakh, MD_{bys}}, {Kazakh, MD_{wbys_a}}, {Kazakh, MD_{wbys_d}}, {Telugu, MD_{wmv_a}}, {Telugu, MD_{wmv_d}} and {Turkish, MP_{wmv}}, where the {Telugu, MD_{wmv_a}} pair experiences the highest absolute performance increase of 0.7%.

6.4.6 Performance on Open-Class Tags

Table 6.9 reports the average precision, recall and F1-score for nouns, verbs and adjectives per target language, across the source languages, in the single-source word-based and stem-based approaches. For complete results per target-source language pair, see Table 2.4 in Appendix B.

[5] We examined the use of n-gram affixes versus affixes generated through segmentation. The former consistently results in better performance.

Target Language	Approach	Noun			Verb			Adjective		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Amharic	Word-Based	67.2	81.5	73.5	75.2	**79.0**	76.6	21.0	6.1	8.1
	Stem-Based	**69.6**	**88.3**	**77.8**	**82.9**	76.9	**79.7**	**48.4**	**25.3**	**32.2**
Basque	Word-Based	58.7	77.6	66.7	50.0	**77.6**	60.2	30.0	10.3	14.6
	Stem-Based	**59.7**	**84.1**	**69.7**	**70.6**	69.0	**69.4**	**36.6**	**21.9**	**27.1**
Finnish	Word-Based	72.1	87.1	78.4	67.9	**82.5**	74.1	**70.3**	32.6	43.2
	Stem-Based	**74.7**	**89.6**	**81.2**	**75.8**	81.5	**78.4**	64.1	**44.6**	**52.2**
Georgian	Word-Based	75.4	88.1	80.8	69.7	**95.5**	80.5	83.9	55.0	64.9
	Stem-Based	**76.7**	**89.2**	**82.4**	**77.6**	95.0	**85.4**	**85.4**	**58.0**	**69.0**
Indonesian	Word-Based	67.2	90.3	76.7	79.8	**87.3**	83.1	48.0	26.1	33.0
	Stem-Based	**68.1**	**91.2**	**77.8**	**84.2**	82.9	**83.4**	**59.4**	**36.4**	**44.9**
Kazakh	Word-Based	65.6	73.8	68.5	47.5	**87.1**	60.7	31.3	4.9	7.9
	Stem-Based	**69.6**	**88.6**	**77.9**	**67.4**	80.6	**73.3**	**67.7**	**18.9**	**29.2**
Telugu	Word-Based	61.2	47.1	53.0	54.7	**96.9**	69.6	**1.9**	**2.2**	**2.0**
	Stem-Based	**64.0**	**60.0**	**61.2**	**65.9**	93.1	**76.8**	0.7	1.1	0.9
Turkish	Word-Based	68.3	73.7	70.2	64.4	**89.2**	74.4	67.3	15.0	22.8
	Stem-Based	**73.6**	**83.6**	**78.2**	**80.7**	80.2	**80.3**	**73.3**	**26.6**	**38.9**

Table 6.9: The average precision, recall and F1-score for nouns, verbs and adjectives per target language, across the source languages, in the single-source word-based and stem-based approaches. The best result per target language and POS tag for each evaluation metric is in **bold.**

In the case of nouns, the stem-based approach improves precision, recall and F1-score consistently for all the languages with an average relative increase in F1-score of 15.7%. This increases to 19.9% in the case of verbs, where the precision and F1-score in the stem-based approach are consistently higher than those in the word-based approach, while the recall in the word-based approach is consistently higher than the one in the stem-based approach. Similarly, in the case of adjectives, the stem-based approach results in an average relative increase in F1-score of 16.1%.

Tables 6.10 and 6.11 report the precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source projection setup MP_{wmv} and the best multi-source decoding setup MD_{bys}, respectively, in the multi-source word-based and stem-based approaches.

The performance of the multi-source setups has a similar pattern to that of the single-source setup, where the stem-based approach outperforms the word-based one but with relatively smaller

Target Language	Approach	Noun			Verb			Adjective		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Amharic	Word-Based	**74.1**	77.9	75.9	73.4	**87.8**	80.0	**55.2**	29.3	38.3
	Stem-Based	73.2	**88.2**	**80.0**	**82.7**	80.7	**81.7**	46.4	**35.1**	**39.8**
Basque	Word-Based	62.1	78.7	69.4	51.9	**76.0**	61.7	36.7	17.7	23.8
	Stem-Based	**62.8**	**83.6**	**71.7**	**72.9**	70.6	**71.7**	**39.1**	**24.0**	**29.6**
Finnish	Word-Based	76.5	88.2	81.9	71.9	**85.1**	78.0	**80.4**	42.0	55.2
	Stem-Based	**78.6**	**89.3**	**83.6**	**78.8**	83.4	**81.0**	67.8	**52.4**	**59.1**
Georgian	Word-Based	**79.1**	**90.6**	**84.5**	73.9	95.5	83.3	**87.7**	**62.8**	**73.1**
	Stem-Based	**79.1**	90.4	84.4	**77.4**	**96.9**	**86.1**	84.3	59.9	70.0
Indonesian	Word-Based	**69.8**	90.2	78.7	85.3	**84.5**	**84.9**	**66.8**	**42.5**	**51.9**
	Stem-Based	**69.8**	**90.8**	**78.9**	**86.2**	82.1	84.1	63.8	40.6	49.6
Kazakh	Word-Based	69.6	75.5	72.5	49.4	**89.5**	63.6	67.4	8.2	14.6
	Stem-Based	**73.1**	**88.5**	**80.0**	**69.7**	81.1	**75.0**	**71.1**	**22.2**	**33.8**
Telugu	Word-Based	**70.6**	46.4	56.0	53.7	**97.3**	69.2	0.0	0.0	0.0
	Stem-Based	70.1	**62.6**	**66.1**	**64.3**	91.7	**75.6**	0.0	0.0	0.0
Turkish	Word-Based	**77.1**	73.2	75.1	66.2	**91.5**	76.8	**82.5**	**31.2**	**45.3**
	Stem-Based	76.6	**82.7**	**79.6**	**80.0**	77.7	**78.9**	80.5	29.3	43.0

Table 6.10: The precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source projection setup, both word-based and stem-based. The best result per target language and POS tag for each evaluation metric is in **bold.**

gaps. Considering the 72 evaluation points (8 target languages, 3 POS tags and 3 evaluation metrics), MP_{wmv} results in the same behavior (in terms of whether the stem-based approach outperforms the word-based one) as that of the single-source setup in 53 evaluation points. This number increases to 65 in the case of MD_{bys}. However, the majority of the differences between the single-source setup and the multi-source ones occur in the case of adjectives.

In the case of MP_{wmv}, the average relative increases in F1-score for nouns, verbs and adjectives in the stem-based approach are 12.8%, 15.0% and 2.6%, respectively, while in the case of MD_{bys}, the average relative increases in F1-score for nouns, verbs and adjectives are 11.7%, 15.2% and 11.4%, respectively. This means that in the multi-source setups, verbs benefit the most from the stem-based approach, followed by nouns and adjectives, in order, while in the single-source setup, adjectives benefit more than nouns from the stem-based approach.

Target Language	Approach	Noun			Verb			Adjective		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Amharic	Word-Based	67.6	84.7	75.2	79.7	77.2	78.4	52.1	6.9	12.2
	Stem-Based	**71.1**	**88.6**	**78.9**	**84.5**	**78.4**	**81.3**	**55.7**	**26.7**	**36.0**
Basque	Word-Based	**63.0**	77.7	69.6	54.2	**75.3**	63.0	36.2	18.3	24.3
	Stem-Based	62.8	**83.4**	**71.7**	**76.3**	68.2	**72.0**	**38.1**	**24.9**	**30.1**
Finnish	Word-Based	78.1	86.3	82.0	72.6	**84.6**	78.2	**68.4**	47.0	55.7
	Stem-Based	**80.4**	**88.6**	**84.3**	**80.0**	83.0	**81.5**	64.9	**58.3**	**61.4**
Georgian	Word-Based	79.2	88.0	83.3	75.0	**97.6**	84.8	85.7	**64.1**	73.3
	Stem-Based	**80.2**	**89.4**	**84.6**	**77.1**	96.7	**85.8**	**89.6**	62.5	**73.6**
Indonesian	Word-Based	**72.3**	89.6	**80.0**	84.9	**86.3**	85.6	64.8	**40.9**	50.1
	Stem-Based	71.4	**90.9**	**80.0**	**88.0**	85.2	**86.6**	**68.5**	40.8	**51.1**
Kazakh	Word-Based	68.1	79.7	73.4	51.6	**88.8**	65.3	70.4	6.2	11.4
	Stem-Based	**72.9**	**88.5**	**79.9**	**71.3**	80.2	**75.5**	**70.9**	**23.7**	**35.5**
Telugu	Word-Based	63.3	55.4	59.1	57.3	**98.3**	72.4	0.0	0.0	0.0
	Stem-Based	**68.2**	**60.2**	**64.0**	**64.5**	94.2	**76.5**	0.0	0.0	0.0
Turkish	Word-Based	75.8	75.9	75.9	69.7	**89.6**	78.4	**84.1**	25.7	39.4
	Stem-Based	**77.2**	**82.3**	**79.6**	**81.8**	79.7	**80.7**	81.1	**28.7**	**42.3**

Table 6.11: The precision, recall and F1-score for nouns, verbs and adjectives per target language in the best multi-source decoding setup, both word-based and stem-based. The best result per target language and POS tag for each evaluation metric is in **bold.**

6.5 Conclusion

In this chapter, we performed unsupervised stem-based POS tagging via annotation projection, where the stem is used instead of the word as the core unit of abstraction. This is useful for low-resource languages with rich affixation, where word-level alignment models are sparse due to rich paradigms and translation inconsistencies that hinder the ability of the alignment models to relate words that belong to the same citation form across parallel texts.

In addition to the stem-based approach, we examined morpheme-based alignment and projection for the cases where both the source and target languages are morphologically complex. Moreover, we experimented with the use of linguistic priors to perform semi-supervised morphological segmentation towards better segmentation and tagging models. Finally, we examined the use of the

generated stems and affixes as learning features in our neural architecture for POS tagging.

We showed the efficiency of the stem-based approach, where it increases the number of training instances and decreases the percentage of OVVs for all the target languages. When evaluated on 48 language pairs of diverse typologies, the stem-based approach outperforms the word-based one in 43 pairs with an average relative error reduction up to 21.4% in the case of Kazakh. We also illustrated that the stem-based approach outperforms the word-based one that operates on three-times more data in about two thirds of the experimental language pairs and thus compensates for the lack of sufficient parallel data. Moreover, we showed that the stem-based approach yields improvements when coupled with multi-source projection and decoding.

When using Arabic as the source language, the morpheme-based approach results in further improvements for all the target languages except Amharic, where Telugu benefits the most. On another hand, using Georgian as a case study, we showed that improving segmentation quality through the use of linguistic priors, in the form of linguist-provided affixes, results in improved POS-tagging models. However, the improvements are not statistically significant for the majority of the experimental language pairs as the improved quality of the generated stems is dominated by the ability of the pipeline to produce high-quality alignments, projections and training instances. In addition, we showed that the use of the generated stems and affixes as learning features in our neural architecture does not yield noticeable improvements since such features are surpassed by the prefix and suffix n-gram character-based features.

Finally, we illustrated that the stem-based approach improves the tagging of open-class words in both the single-source and multi-source setups.

Chapter 7

Conclusion and Future Directions

> "Look at a day when you are supremely satisfied at the end. It's not a day when you lounge around doing nothing, it's when you've had everything to do and you've done it. " — *Margaret Thatcher*

7.1 Summary of Contributions

In this thesis, we have proposed several contributions and accompanying findings in two morphology tasks: unsupervised morphological segmentation and unsupervised cross-lingual part-of-speech (POS) tagging. Our empirical results show significant improvements over previously state-of-the-art systems, moving towards reducing the gaps to the corresponding supervised models.

7.1.1 Two Morphology Systems

We introduced an unsupervised and minimally supervised morphological-segmentation framework, *MorphAGram*, that is based on Adaptor Grammars (AGs) and allows for the inclusion of linguistic priors (Section 7.1.2). In *MorphAGram*, we defined several language-independent grammars of different characteristics. In addition, we introduced a fully unsupervised learning setting that relies on self-training through two rounds of learning in order to approximate the effect of utilizing scholar knowledge. Moreover, since there is no single grammar that works best across all languages, we developed an approach to automatically select a nearly optimal configuration for unseen languages.

We also introduced an end-to-end approach for unsupervised cross-lingual POS tagging via

annotation projection in truly low-resource scenarios, where we do not assume access to large or domain-specific parallel data. Our approach exploits and expands the best practices in the literature in order to produce high-quality projected annotations towards highly efficient POS models. As part of our approach, we developed a rich neural architecture that combines non-contextualized and transformer-based contextualized word embeddings along with affix embeddings and word-cluster embeddings.

We finally combined our work on unsupervised morphological segmentation and unsupervised cross-lingual POS tagging by introducing an approach for unsupervised stem-based cross-lingual POS tagging via annotation projection, where the stem is the core unit of abstraction for alignment and projection, which is beneficial to low-resource languages of rich morphology. Moreover, we examined morpheme-based alignment and projection and the use of segmentation information (stems and affixes) as learning features in our neural architecture.

7.1.2 Incorporation of Linguistic Priors

In addition to the fully unsupervised language-independent settings, *MorphAGram* can benefit from scholar knowledge in the form of affixes seeded into the grammars, where it handles two different cases where the scholar knowledge is either generated from grammar references as weak linguistic priors or compiled by an expert in the underlying language as strong linguistic priors. In addition, we introduced another method for the incorporation of linguistic priors in the form of a grammar definition through the design of a language-specific grammar. We also examined the use of linguistic priors towards better POS models in stem-based cross-lingual POS tagging.

7.1.3 Multilingual and Multi-Source Learning

In the case of unsupervised morphological segmentation, we examined multilingual setups in which we combine the lexicons of multiple related languages within low-resource setups. On another hand, we introduced eight multi-source cross-lingual POS-tagging setups that make use of multiple source languages, as parallel data might be available between the target language and multiple source ones,

either in the projection phase or at decoding. Our multi-source approaches are based on weighted maximum voting and Bayesian inference, in addition to hybrid setups in which we combine the two mechanisms.

7.1.4 Evaluation and Analysis

We conducted comprehensive evaluation and analysis for our frameworks. We evaluated our morphological-segmentation framework *MorphAGram* on 13 languages of diverse typologies: analytic (English), fusional (German and Arabic) agglutinative (Turkish, Finnish, Estonian, Zulu and Japanese) and synthetic/polysynthetic (Georgian, Mexicanero, Nahuatl (Mexicano), Wixarika (Huichol) and Mayo (Yorem Nokki)). We showed that our fully unsupervised system achieves an average F1-score of 75.4%, using the BPR metric, and outperforms both *Morfessor* (Creutz and Lagus, 2007; Grönroos et al., 2014) and *MorphoChain* (Narasimhan et al., 2014), two state-of-the-art baselines, with average relative error reductions of 22.8% and 40.7%, respectively. We also illustrated the benefits of incorporating linguistic priors by achieving noticeable relative error reductions for Japanese, Georgian and Arabic, as case studies. We also showed performance gains for Estonian upon combining small Finnish and Estonian lexicons. In addition, we analyzed the morphological characteristics, the performance across datasets of various sizes and the segmentation output for each experimental language.

We evaluated our POS-tagging framework on six source languages, namely English, Spanish, French, German, Russian and Arabic, and 14 target languages of diverse typologies, namely Afrikaans, Amharic, Basque, Bulgarian, Finnish, Georgian, Hindi, Indonesian, Kazakh, Lithuanian, Persian, Portuguese, Telugu and Turkish, for a total of 84 language pairs. Our system achieves an average POS accuracy of 75.5% across all the language pairs, where we get the best results when transferring across related languages.

We also showed that the multi-source setups outperform the single-source one on average and in 10 target languages. We also illustrated the efficiency of the stem-based approach, where it outperforms the word-based approach in 43 language pairs out of 48 experimental ones with an

average relative error reduction up to 21.4% in the case of Kazakh. Moreover, the stem-based approach outperforms the word-based one that operates on three-times more data in about two thirds of the experimental language pairs and thus compensates for the lack of sufficient parallel data. The stem-based approach also yields further improvements when coupled with multi-source projection and decoding. We also showed improvements using morpheme-based alignment and projection using Arabic as the source language and improvements using linguistic priors for stem-based alignment and projection using Georgian as the target language. We moreover illustrated the efficiency of our approaches at analyzing open-class words.

In addition, we showed significant improvements over two state-of-the-art unsupervised systems by Buys and Botha (2016) and Agić et al. (2016) and two state-of-the-art semi-supervised systems by Cotterell and Heigold (2017) and Plank et al. (2016) despite the fact that the systems we compare to use either large and/or domain-specific parallel data, several source languages, some labeled data or language resources.

In addition, we analyzed two ablation setups that prove the efficiency of our approaches when lacking large monolingual data and/or rich computational resources. We also demonstrated that unsupervised cross-lingual POS tagging via annotation projection might be an alternative to supervised learning, where it predicts at least as many as 85.0% of the correct decisions made by the state-of-the-art supervised system *Stanza* (Qi et al., 2020) in eight target languages. Finally, we illustrated that annotation projection is less sensitive to the relatedness between the source and target languages when compared to zero-shot model transfer (Pires et al., 2019).

7.1.5 New Language Resources

As part of our evaluation, we developed two gold-standard morphological-segmentation datasets for Japanese and Georgian, each containing 1,000 words and their gold segmentation, where a word might receive more than one possible analysis. In addition, the Georgian dataset contains the main POS category of each word. We also developed a gold-standard POS-labeled dataset for Georgian of 1,000 sentences, where the tags follow the Universal-Dependencies tagging scheme.

7.2 Future Directions

In this section, we discuss possible research directions that would further enhance unsupervised morphological segmentation and unsupervised cross-lingual POS tagging and their downstream applications.

7.2.1 Introducing PSRCGs into MorphAGram

The current PCFGs are not able to handle several morphological phenomena such as infixation, circumfixation and root-templatic derivation. This was the motivation for Botha and Blunsom () to extended AGs by replacing PCFGs by PSRCGs (probabilistic simple-range concatenating grammars) for the processing of Arabic and Hebrew, two morphologically complex languages of templatic morphology. In a PSRCG, nonterminals accept arguments (variables), where a nonterminal becomes instantiated when the variables are bound to ranges through substitution. We hypothesize that the introduction of PSRCGs into our morphological-segmentation framework *MorphAGram* would be beneficial to better model languages of complex morphological phenomena, while exploiting the capabilities of *MorphAGram*, which include, but are not limited to, accessing several language-independent grammars within different learning settings, the automatic handling of unseen languages and the incorporation of linguistic priors.

7.2.2 Linguistic Priors for Multilingual Morphological Segmentation

We showed that the incorporation of linguistic priors, either in the form of a grammar definition, in the case of Japanese, or linguist-provided affixes, in the case of Georgian and Arabic, improves the performance of morphological segmentation by noticeable relative BPR error reductions of 4.2%, 33.2% and 32.9% for Japanese, Georgian and Arabic, respectively (Sections 3.3.4 and 3.5.5). On another hand, we demonstrated that multilingual training, in which we combine the lexicons of multiple related languages, helps in some of the low-resource setups, where combining small Finnish and Estonian lexicons of 500 and 1,000 words improves the morphological-segmentation

performance for Estonian by relative BPR error reductions of 2.5% and 4.5% (Sections 3.3.5 and 3.5.6), respectively. Accordingly, we hypothesize that coupling linguistic priors with multilingual learning would help the sampler better derive a segmentation model that generalizes well across the underlying languages, especially in low-resource learning setups. This can be achieved either by tailoring a grammar that models the linguistic characteristics of a specific language family or genus in which the languages share the main aspects of word structure or by seeding linguist-provided affixes that are common across a group of related languages of highly overlapping sets of affixes.

7.2.3 Morphologically Driven Tokenization in Neural-Based NLP Tasks

Subword-based tokenization has become a popular choice in several neural-based NLP tasks, such as neural machine translation (Artetxe et al., ; Bawden et al., 2019) and building transformer-based language models, e.g., *BERT* (Devlin et al., 2019), *XLNet* (Yang et al., 2019), *RoBERTa* (Conneau et al., 2019), *GPT-2* (Radford et al., 2019), GPT-3 (Brown et al., 2020), *ALBERT* (Lan et al., 2020) and *ELECTRA* (Clark et al., 2020). There are currently three widely used subword-based tokenization methods.

- *WordPiece* (Yonghui et al., 2016): In *WordPiece*, the vocabulary is first initialized with the characters in the underlying language(s), and then the most frequent character sequences are iteratively added to the vocabulary. *WordPiece* is used in *BERT*, *ALBERT* and *ELECTRA*.

- *BPE* (byte-pair encoding) (Gage, 1994; Sennrich et al., 2016): *BPE* relies on data compression in which the most frequent pair of consecutive bytes in the data is replaced by a newly introduced byte, and then the process repeats iteratively for a preset number of iterations. *BPE* is used in *GPT-2* and *GPT-3*.

- *SentencePiece* (Kudo and Richardson, 2018): *SentencePiece* relies on *BPE*, coupled with the word-based unigram tokenization method proposed by (Kudo, 2018). *SentencePiece* is used in *RoBERTa* and *XLNet*.

One path to investigate is the use of morphologically motivated tokenization in the recent neural-based NLP tasks, such as neural machine translation and building transformer-based language models. The high cost of morphological segmentation has motivated the use of cheaper non-linguistically driven tokenization schemes that rely on relatively simple statistical methods, especially with the need to obtain expensive labeled data in order to produce high-quality morphemes, which is not appropriate for processing low-resource languages and for developing multilingual models, such as *mBERT* and *XLM-R*. However, the recent advances in unsupervised morphological segmentation, currently led by our state-of-the-art morphological-segmentation framework *MorphAGram*, opens the door to harnessing morphologically motivated tokenization in several neural applications. Pan et al. (2020) showed promising results, where the use of the stems and suffixes as the core unit of abstraction in neural machine translation outperforms *BPE*, while the best results are obtained by using *BPE* on top of the stems and suffixes.

7.2.4 The Role of Morphological Typology in Cross-Lingual Learning

As demonstrated in Section 4.4.2, languages that belong to the same family transfer best across each other. For instance, English and German are the best sources for Afrikaans (IE, Germanic), while Spanish transfers best to Portuguese (IE, Romance), and Russian yields the best performance for Bulgarian (IE, Slavic). This indicates that the selection of the source language is crucial for the processing of the language of interest, where the typological similarities between the source and the target have a direct impact on the performance of the ultimate POS model.

On the alignment side, one main aspect is that word-level alignment models tend to learn better across languages of similar word order, such as Subject-Verb-Object order and Adjective-Noun order. On the projection side, transferring POS suffers across languages of different POS assumptions. For instance, it's argued that Korean does not technically contain adjectives but rather expresses properties of nouns via stative verbs (Kim, 2002). As a result, transferring from English, for instance, to Korean would result in undesirable adjectival tags. This problem is more obvious when transferring other morphosyntactic features such as gender and case, where transferring

from English to Arabic, for instance, would fail to assign gender labels to the adjectives in Arabic and leave case endings unrecognized. Therefore, failing to select a source language that adopts morphological assumptions that are similar to those of the target language would weaken the quality of annotation projection.

We believe that developing disciplined guidelines for the selection of an appropriate source language to transfer from should enhance cross-lingual learning. This can be achieved by studying what typological features are highly significant for the alignment and projection phases with respect to the underlying task. Furthermore, the research in that direction would further motivate the unsupervised learning of typological features for the processing of low-resource languages whose description is inadequate or lacking.

Bibliography

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics and the 2015 International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 268–272.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics (TACL)*, 4:301–312.

Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the 2013 Conference on Computational Natural Language Learning (CoNLL)*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Ebrahim Ansari, Zdeněk Žabokrtskỳ, Mohammad Mahmoudi, Hamid Haghdoost, and Jonáš Vidra. 2019. Supervised morphological segmentation using rich annotated lexicon. In *Proceedings of the 2019 International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 52–61.

Howard I Aronson. 1990. *Georgian: A Reading Grammar, Corrected Edition*. Slavica Publishers.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *Proceedings of the 2018 International Conference on Learning Representations (ICLR)*.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, 108.

Michele Banko and Robert C Moore. 2004. Part of speech tagging in context. In *Proceedings of the 2004 International Conference on Computational Linguistics (COLING)*, page 556. Association for Computational Linguistics.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on*

Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 103–115, Florence, Italy. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Jan A Botha and Phil Blunsom. Adaptor grammars for learning non-concatenative morphology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the 2000 Conference on on Applied Natural Language Processing (ANLP)*, pages 224–231. Association for Computational Linguistics.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 2020 conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 1954–1964, Berlin, Germany.

Christos Christodouloupoulos and Mark Steedman. 2014. A massively parallel corpus: the Bible in 100 languages. In *Proceedings of the 2014 International Conference on Language Resources and Evaluation (LREC)*, volume 49, pages 375–395. Springer.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Proceedings of Human Language*

Technologies: The 2020 Annual Conference of the Association for Computational Linguistics, (ACL).

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the 2002 ACL Workshop on Morphological and Phonological Learning*, volume 6, pages 21–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2004. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 43–51.

Mathias Creutz and Krista Lagus. 2005a. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the 2005 International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*, volume 1, pages 51–59.

Mathias Creutz and Krista Lagus. 2005b. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology Helsinki.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–34.

Mathias Creutz. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of Human Language Technologies: The 2003 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 280–287.

Silviu Cucerzan and David Yarowsky. 2002. Bootstrapping a multilingual part-of-speech tagger in one person-day. Technical report, John Hopkins Univ Baltimore Md Center For Language And Speech Processing (CLSP).

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of Human Language Technologies: The 2011 Annual*

Conference of the Association for Computational Linguistics, (ACL), pages 600–609. Association for Computational Linguistics.

Hervé Déjean. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In *New Methods in Language Processing and Computational Natural Language Learning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Human Language Technologies: The 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013. Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 634–639.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 644–648.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the 2011 International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 118–124.

Micha Elsner, Eugene Charniak, and Mark Johnson. 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 (NAACL)*, pages 164–172.

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

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2018. Automatically tailoring unsupervised

morphological segmentation to the language. In *Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Brussels, Belgium.

Ramy Eskander, Judith L Klavans, and Smaranda Muresan. 2019. Unsupervised morphological segmentation for low-resource polysynthetic languages. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith L Klavans, and Smaranda Muresan. 2020a. Morphagram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the 2020 International Conference on Language Resources and Evaluation (LREC)*, pages 7112–7122.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020b. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ramy Eskander, Cass Lowry, Sujay Khandagale, Francesca Callejas, Judith Klavans, Maria Polinsky, and Smaranda Muresan. 2021. Minimally-supervised morphological segmentation using adaptor grammars with linguistic priors. In *Proceedings of Human Language Technologies: The 2021 Annual Conference of the Association for Computational Linguistics, ACL Findings, (ACL)*.

Meng Fang and Trevor Cohn. 2016. Learning when to trust distant supervision: An application to low-resource POS tagging using cross-lingual projection. In *Proceedings of the 2016 Conference on Computational Natural Language Learning (CoNLL)*, pages 178–186, Berlin, Germany. Association for Computational Linguistics.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Proceedings of the 2005 International Conference on Natural Language Processing (ICON)*, pages 862–873. Springer.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

John Goldsmith and Yu Hu. 2004. From signatures to finite state automata. In *Midwest Computational Linguistics Colloquium*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Barbara B Greene and Gerald M Rubin. 1971. *Automatic grammatical tagging of English*. Department of Linguistics, Brown University.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of the 2014 International Conference on Computational Linguistics (COLING)*, pages 1177–1185.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of Human Language Technologies: The 2005 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 573–580.

Zellig S Harris. 1970. Morpheme boundaries within words: Report on a computer test. In *Papers in Structural and Transformational Linguistics*, pages 68–77. Springer.

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

Yun Huang, Min Zhang, and Chew Lim Tan. 2011. Nonparametric Bayesian machine transliteration with synchronous adaptor grammars. In *Proceedings of Human Language Technologies: The 2011 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 534–539.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

L Ron Hubbard and Lloyd Sherr. 2007. *Dianetics: The modern science of mental health*. New Era.

Matthias Huck, Diana Dutka, and Alexander Fraser. 2019. Cross-lingual annotation projection is effective for neural part-of-speech tagging. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 223–233.

Mark Johnson and Katherine Demuth. 2010. Unsupervised phonemic Chinese word segmentation

using adaptor grammars. In *Proceedings of the 2010 International Conference on Computational Linguistics (COLING)*, pages 528–536. Association for Computational Linguistics.

Howard Johnson and Joel Martin. 2003. Unsupervised learning of morphology for English and Inuktitut. In *Proceedings of Human Language Technologies: The 2003 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 43–45.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In B. Sch"olkopf, J. Platt, and T. Hoffman, editors, *Proceedings of the 2007 conference on Advances in Neural Information Processing Systems (NeurIPS)*, pages 641–648, Cambridge, MA. MIT Press.

Mark Johnson. 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, Columbus, Ohio. Association for Computational Linguistics.

Mark Johnson. 2008b. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of Human Language Technologies: The 2008 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 398–406.

Michael Irwin Jordan. 1998. *Learning in graphical models*, volume 89. Springer Science & Business Media.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *Proceedings of Human Language Technologies: The 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 47–57.

Dimitar Kazakov. 1997. Unsupervised learning of naive morphology with genetic algorithms. In *Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks*, pages 105–112.

Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pages 619–627. PMLR.

Min-Joo Kim. 2002. Does Korean have adjectives? *MIT working papers in linguistics*, 43:71–89.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 2014 International Conference on Learning Representations (ICLR)*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, volume 5, pages 79–86. Citeseer.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *Proceedings of the 4th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 145–154.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of Human Language Technologies: The 2018 Annual Conference of the Association for Computational Linguistics, (ACL): ACL Findings*.

Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech & Language*, 6(3):225–242.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 2020 International Conference on Learning Representations (ICLR)*.

Shen Li, Joao V Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Pro-*

cessing and Computational Natural Language Learning (EMNLP-CoNLL), pages 1389–1398. Association for Computational Linguistics.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

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

Alec Marantz. 2001. Words and things. *Handout, MIT*.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The penn treebank.

André FT Martins and Julia Kreutzer. 2017. Learning what's easy: Fully differentiable neural easy-first taggers. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 349–362.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the 2014 International Conference on Language Resources and Evaluation (LREC)*, pages 3158–3163.

Karthik Narasimhan, Damianos Karakos, Richard M. Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of Human Language Technologies: The 2017 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 1470–1480, Vancouver, Canada. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling, Jörg Tiedemann, et al. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-speech tagging for Twitter: Word clusters and other advances. *School of Computer Science, Carnegie Mellon University, Tech. Rep*.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological word segmentation on agglutinative languages for neural machine translation. *arXiv preprint arXiv:2001.01589*.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the 2014 International Conference on Language Resources and Evaluation (LREC)*, volume 14, pages 1094–1101.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the 2011 International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In

Proceedings of Human Language Technologies: The 2019 Annual Conference of the Association for Computational Linguistics, (ACL), pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Jim Pitman. 1995. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158.

Barbara Plank and Željko Agić. 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 614–620, Brussels, Belgium. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

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

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the Association for Computational Linguistics, (ACL): System Demonstrations*, pages 101–108, Online, July. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of Human Language Technologies: The 2019 Annual Conference of the Association for Computational Linguistics, (ACL)*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 328–338.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Christian Robert and George Casella. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.

JK Rowling. 1997. *Harry Potter and the Sorcerer's Stone: The Illustrated Edition (Harry Potter)*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the 2013 Conference on Computational Natural Language Learning (CoNLL)*, pages 29–37.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. A comparative study of minimally supervised morphological segmentation. *Computational Linguistics*, 42(1):91–120.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the Association for Computational Linguistics, ACL Findings, (ACL)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

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

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, Mikko Kurimo, et al. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of Human Language Technologies: The 2014 Annual Conference of the Association for Computational Linguistics, (ACL)*. Aalto University.

Sebastian Spiegler and Christian Monson. 2010. EMMA, A novel evaluation metric for morphological analysis. In *Proceedings of the 2010 International Conference on Computational Linguistics (COLING)*, pages 1029–1037, Beijing, China.

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

Maria Sukhareva, Francesco Fuscagni, Johannes Daxenberger, Susanne Görke, Doris Prechel, and Iryna Gurevych. 2017. Distantly supervised POS tagging of low-resource languages under extreme data sparsity: The case of Hittite. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 95–104.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics (TACL)*, 1:1–12.

Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of the 2014 International Conference on Computational Linguistics (COLING)*, pages 1854–1864.

Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor baseline.

Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. 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 (EMNLP-CoNLL)*, pages 699–709. Association for Computational Linguistics.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 2018 International Conference on Computational Linguistics (COLING)*, pages 3879–3889, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 2019 conference on Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the 2001 International Conference on Human language Technology Research (HLT)*, pages 1–8. Association for Computational Linguistics.

W Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Daniel Zeman, Martin Popel, Milan Straka, and et al. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Appendix A: Unsupervised Morphological Segmentation

Tables 1.1 and 1.2 report the performance of our *MorphAGram* framework using the BPR and EMMA-2 metrics, respectively, for the test languages, namely Japanese, Georgian, Arabic, Mexicanero, Nahuatl, Wixarika and Mayo, using the nine grammars defined in Section 3.3.4.2.

Language	Grammar	*Standard* Setting			*Cascaded* Setting			*Scholar-Seeded* Setting		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Japanese	*Morph+SM*	86.5	62.8	72.7	86.7	62.8	72.8	87.3	63.2	73.3
	Simple	80.7	74.8	77.6	80.8	74.9	77.7	80.9	75.0	77.8
	Simple+SM	83.0	72.9	77.6	83.1	72.7	77.5	83.0	72.7	77.5
	PrStSu	77.9	81.5	79.7	78.2	81.2	79.6	79.3	80.7	**80.0**
	PrStSu+SM	81.7	77.9	**79.8**	81.5	78.2	**79.8**	82.3	77.6	79.9
	PrStSu+Co+SM	95.3	40.0	56.3	94.3	44.4	60.4	95.6	39.7	56.1
	PrStSu2a+SM	79.4	79.1	79.2	78.8	80.0	79.4	79.8	80.3	**80.0**
	PrStSu2b+SM	75.7	78.6	77.1	75.7	79.0	77.3	75.4	78.7	77.0
	PrStSu2b+Co+SM	99.8	23.3	37.8	99.7	24.7	39.6	99.7	25.3	40.3
Georgian	*Morph+SM*	85.6	52.8	65.3	85.5	52.3	64.9	85.6	52.3	64.9
	Simple	75.2	63.9	69.1	75.4	64.1	69.3	75.6	64.3	69.5
	Simple+SM	81.6	60.9	69.7	81.4	60.6	69.5	81.9	61.1	70.0
	PrStSu	67.3	70.0	68.6	77.0	75.4	76.2	79.6	72.5	**75.9**
	PrStSu+SM	82.0	69.1	**75.0**	82.9	71.7	**76.9**	84.3	67.9	75.2
	PrStSu+Co+SM	87.7	50.4	64.0	87.9	50.1	63.9	88.0	51.1	64.7
	PrStSu2a+SM	78.0	70.3	74.0	81.8	64.2	72.0	83.0	62.0	71.0
	PrStSu2b+SM	71.7	68.5	70.1	72.3	69.7	71.0	72.7	69.2	70.9
	PrStSu2b+Co+SM	99.8	13.1	23.2	99.9	13.2	23.3	99.7	13.4	23.6
Arabic	*Morph+SM*	85.9	66.6	75.0	86.0	66.3	74.9	86.3	67.2	75.6
	Simple	64.2	75.6	69.4	63.9	75.2	69.1	64.6	75.7	69.7
	Simple+SM	76.7	82.0	79.3	76.0	81.8	78.8	76.4	82.1	79.1
	PrStSu	65.3	86.4	74.4	68.5	85.3	76.0	68.3	86.0	76.1
	PrStSu+SM	77.5	88.2	**82.5**	76.3	86.4	**81.1**	76.8	88.9	**82.4**
	PrStSu+Co+SM	87.7	63.9	73.9	87.4	64.9	74.5	87.5	64.7	74.3
	PrStSu2a+SM	74.6	82.8	78.5	75.0	83.2	78.9	72.8	81.9	77.1
	PrStSu2b+SM	77.4	82.9	80.0	77.5	82.8	80.1	75.1	81.9	78.4
	PrStSu2b+Co+SM	99.9	19.9	33.2	100.0	19.6	32.8	99.9	19.6	32.8

	Morph+SM	81.3	72.1	76.4	81.1	72.6	76.6	81.3	72.5	76.6
	Simple	69.0	69.0	69.0	71.0	71.4	71.2	70.5	69.2	69.8
	Simple+SM	79.0	72.8	75.8	78.2	71.7	74.8	78.1	71.9	74.9
	PrStSu	69.4	84.9	76.3	69.9	79.8	74.5	72.3	86.9	78.9
Mexicanero	*PrStSu+SM*	77.9	81.0	**79.4**	77.3	77.5	77.4	82.9	82.1	**82.5**
	PrStSu+Co+SM	84.0	67.0	74.5	79.3	71.4	75.1	82.2	70.1	75.6
	PrStSu2a+SM	74.1	74.1	74.1	76.0	76.8	76.4	76.4	78.2	77.2
	PrStSu2b+SM	77.1	72.4	74.7	78.9	77.2	**78.0**	80.4	76.6	78.4
	PrStSu2b+Co+SM	100.0	45.8	62.9	99.2	45.9	62.7	99.8	46.0	62.9
	Morph+SM	66.6	65.1	65.8	65.6	66.7	66.1	67.5	66.4	66.9
	Simple	50.6	61.8	55.6	50.4	62.1	55.6	50.9	62.6	56.2
	Simple+SM	62.2	66.7	64.4	62.9	66.7	64.7	62.0	65.4	63.6
	PrStSu	50.1	81.8	62.1	51.8	78.2	62.3	53.1	81.7	64.3
Nahuatl	*PrStSu+SM*	60.8	74.6	**67.0**	62.9	71.8	**67.0**	63.3	76.1	**69.1**
	PrStSu+Co+SM	71.0	60.3	65.2	67.1	60.6	63.6	69.8	63.7	66.6
	PrStSu2a+SM	58.0	72.4	64.4	61.2	71.7	66.0	58.8	73.7	65.4
	PrStSu2b+SM	59.3	70.0	64.1	61.5	70.8	65.8	62.4	70.9	66.4
	PrStSu2b+Co+SM	99.8	33.8	50.5	98.0	34.0	50.5	99.7	34.0	50.7
	Morph+SM	85.4	55.6	67.4	84.2	56.8	67.8	83.6	56.2	67.2
	Simple	70.3	59.4	64.4	69.2	58.8	63.5	69.8	59.8	64.4
	Simple+SM	81.5	58.3	68.0	81.0	57.3	67.1	82.3	58.3	68.2
	PrStSu	66.2	84.6	74.3	70.7	79.9	75.0	67.1	85.0	75.0
Wixarika	*PrStSu+SM*	82.7	70.9	**76.4**	82.9	70.2	**76.0**	81.1	74.9	**77.9**
	PrStSu+Co+SM	88.0	49.0	63.0	85.6	54.5	66.6	84.5	56.2	67.5
	PrStSu2a+SM	74.4	67.2	70.6	78.1	70.8	74.3	75.4	75.0	75.2
	PrStSu2b+SM	79.0	66.8	72.4	81.2	68.0	74.0	79.5	72.1	75.6
	PrStSu2b+Co+SM	98.7	17.7	30.0	98.8	17.4	29.6	97.7	20.2	33.4
	Morph+SM	85.9	67.2	75.4	84.9	67.3	75.1	84.2	65.1	73.5
	Simple	65.2	68.8	66.9	64.9	69.8	67.3	67.5	71.6	69.4
	Simple+SM	80.0	70.4	74.8	77.1	71.3	74.1	79.1	73.3	76.1
	PrStSu	59.8	87.8	71.1	62.8	87.4	73.0	63.8	88.0	74.0
Mayo	*PrStSu+SM*	78.4	79.6	**78.8**	82.9	78.8	**80.8**	84.4	78.7	**81.5**
	PrStSu+Co+SM	88.0	60.4	71.6	85.6	60.4	70.8	87.2	63.9	73.8
	PrStSu2a+SM	80.0	75.8	77.8	81.7	74.4	77.9	82.0	75.0	78.4
	PrStSu2b+SM	67.4	77.3	72.0	63.6	79.2	70.5	66.2	78.1	71.6
	PrStSu2b+Co+SM	98.4	24.0	38.6	98.4	24.2	38.9	97.6	24.0	38.6

Table 1.1: The segmentation performance (BPR) of the different grammars on the test languages. The best result per language-setting pair is in **bold**. The best language-independent result per language is underlined.

Language	Grammar	*Standard* Setting			*Cascaded* Setting			*Scholar-Seeded* Setting		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Japanese	*Morph+SM*	95.8	70.5	81.2	95.6	70.2	81.0	95.9	70.6	81.3
	Simple	92.0	78.0	84.4	92.2	78.5	84.8	92.1	78.5	84.8
	Simple+SM	93.3	76.7	84.2	93.3	76.3	84.0	93.2	76.3	83.9
	PrStSu	87.8	84.1	85.9	88.1	84.5	86.3	88.7	83.5	86.0
	PrStSu+SM	91.0	81.9	**86.2**	91.0	82.4	**86.5**	91.5	81.4	**86.1**
	PrStSu+Co+SM	99.2	56.8	72.2	98.7	60.3	74.8	99.3	56.9	72.4
	PrStSu2a+SM	89.6	82.0	85.6	89.1	82.8	85.9	89.5	82.9	**86.1**
	PrStSu2b+SM	87.9	81.0	84.3	88.3	80.9	84.5	88.1	80.6	84.1
	PrStSu2b+Co+SM	99.9	47.7	64.5	100.0	49.2	65.9	100.0	49.7	66.4
Georgian	*Morph+SM*	92.5	54.1	68.3	92.5	53.4	67.7	92.7	53.6	67.9
	Simple	87.2	60.1	71.2	86.7	60.3	71.1	86.9	60.5	71.3
	Simple+SM	91.2	58.8	71.5	91.1	59.0	71.6	91.1	59.1	71.7
	PrStSu	78.7	64.7	71.0	84.2	70.6	76.8	87.0	68.2	**76.5**
	PrStSu+SM	88.4	65.9	**75.5**	88.8	67.8	**76.9**	90.0	64.8	75.3
	PrStSu+Co+SM	93.3	53.1	67.7	93.5	52.7	67.4	93.5	53.5	68.0
	PrStSu2a+SM	86.1	65.7	74.5	90.6	61.6	73.3	91.3	59.9	72.4
	PrStSu2b+SM	82.1	63.7	71.7	82.3	64.7	72.4	82.7	64.1	72.2
	PrStSu2b+Co+SM	99.9	32.2	48.7	99.9	32.2	48.7	99.9	32.3	48.9
Arabic	*Morph+SM*	94.3	75.6	83.9	94.1	75.5	83.8	94.3	75.8	84.0
	Simple	88.0	80.3	84.0	88.1	80.2	84.0	87.9	80.3	83.9
	Simple+SM	91.6	83.5	87.4	91.3	83.4	87.2	91.3	83.5	87.2
	PrStSu	82.3	88.6	85.3	84.0	86.6	85.3	83.6	87.6	85.6
	PrStSu+SM	88.1	88.7	**88.4**	88.1	86.8	87.4	87.5	89.4	**88.4**
	PrStSu+Co+SM	94.9	74.3	83.3	95.0	74.7	83.7	94.9	74.5	83.5
	PrStSu2a+SM	90.5	83.8	87.0	90.5	84.1	87.2	90.0	83.4	86.6
	PrStSu2b+SM	91.6	83.9	87.6	91.7	83.8	**87.5**	91.0	83.5	87.1
	PrStSu2b+Co+SM	100.0	50.8	67.4	100.0	50.8	67.3	100.0	50.8	67.4
Mexicanero	*Morph+SM*	93.7	83.1	88.0	93.0	83.3	87.9	93.1	83.2	87.9
	Simple	93.2	82.5	87.5	92.4	82.4	87.1	93.2	82.3	87.4
	Simple+SM	94.2	83.8	88.7	94.8	83.2	**88.6**	93.7	83.0	88.0
	PrStSu	81.3	91.5	86.0	82.2	85.9	84.0	84.0	92.5	88.1
	PrStSu+SM	91.2	89.0	**90.1**	90.1	85.5	87.7	92.3	90.7	**91.5**
	PrStSu+Co+SM	94.8	80.0	86.8	92.7	83.8	88.0	93.5	82.2	87.5
	PrStSu2a+SM	90.9	83.9	87.2	91.2	85.3	88.2	90.9	87.8	89.4
	PrStSu2b+SM	92.7	83.4	87.8	92.2	85.1	88.5	92.3	85.6	88.8
	PrStSu2b+Co+SM	100.0	73.2	84.5	99.8	73.2	84.5	99.8	73.2	84.5

Nahuatl	*Morph+SM*	86.2	77.8	81.7	85.5	79.3	**82.3**	87.2	78.0	82.3
	Simple	81.2	75.9	78.4	81.7	75.3	78.4	81.8	75.1	78.3
	Simple+SM	87.5	77.6	82.2	87.1	76.5	81.5	87.5	77.5	82.2
	PrStSu	63.6	87.6	73.7	65.6	84.9	74.0	66.4	87.9	75.6
	PrStSu+SM	81.4	85.6	**83.4**	82.5	81.6	82.0	81.2	87.5	**84.2**
	PrStSu+Co+SM	89.8	75.7	82.1	87.9	75.4	81.1	88.4	77.3	82.5
	PrStSu2a+SM	78.7	82.3	80.4	82.4	81.4	81.9	79.3	82.2	80.7
	PrStSu2b+SM	80.2	80.3	80.2	82.2	81.1	81.7	82.6	81.1	81.9
	PrStSu2b+Co+SM	100.0	64.3	78.3	100.0	64.3	78.3	100.0	64.3	78.3
Wixarika	*Morph+SM*	92.3	60.3	72.9	89.8	60.7	72.4	90.8	60.9	72.9
	Simple	85.9	63.7	73.1	85.4	63.3	72.7	85.5	63.2	72.7
	Simple+SM	90.6	61.2	73.1	91.2	61.3	73.3	90.7	61.2	73.1
	PrStSu	67.2	86.4	75.6	69.9	80.5	74.8	67.1	87.0	75.8
	PrStSu+SM	85.9	75.7	**80.4**	85.8	73.1	**78.9**	84.7	79.5	**82.0**
	PrStSu+Co+SM	94.8	55.1	69.7	92.4	58.9	71.9	91.2	61.1	73.1
	PrStSu2a+SM	80.5	68.6	74.1	82.0	72.9	77.2	79.5	78.2	78.7
	PrStSu2b+SM	85.4	68.2	75.8	85.4	69.8	76.8	83.4	75.0	79.0
	PrStSu2b+Co+SM	100.0	42.9	60.0	100.0	42.5	59.7	99.6	43.8	60.9
Mayo	*Morph+SM*	94.2	80.5	86.8	92.8	79.8	85.9	93.5	79.4	85.8
	Simple	85.7	80.2	82.9	86.2	80.2	83.1	86.5	81.0	83.7
	Simple+SM	91.8	80.6	85.8	90.3	81.7	85.8	91.6	83.0	87.1
	PrStSu	69.0	90.5	78.3	71.0	89.9	79.4	71.6	89.9	79.7
	PrStSu+SM	88.5	87.7	**88.1**	89.5	87.5	**88.5**	91.8	88.2	**89.9**
	PrStSu+Co+SM	95.8	77.5	85.7	94.4	76.4	84.5	95.2	78.7	86.1
	PrStSu2a+SM	90.5	85.1	87.7	90.3	84.6	87.3	91.1	86.0	88.5
	PrStSu2b+SM	81.8	83.8	82.8	78.3	85.1	81.5	81.0	85.8	83.3
	PrStSu2b+Co+SM	100.0	60.9	75.7	100.0	60.5	75.4	100.0	60.5	75.4

Table 1.2: The segmentation performance (EMMA-2) of the different grammars on the test languages. The best result per language-setting pair is in **bold**. The best language-independent result per language is underlined.

Appendix B: Unsupervised Cross-Lingual Part-of-Speech Tagging

Table 2.1 reports the precision, recall and F1-score for nouns, verbs and adjectives per target-source language pair using our word-based unsupervised cross-lingual POS-tagging system when using the Bible as the source of parallel data and evaluating on the test sets of UD-v2.5. We use only the New Testament in the cases of Basque, Georgian and Kazakh, and we use in-house annotations for Georgian.

Target	Source	Noun			Verb			Adjective		
Language	Language	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Afrikaans	English	90.8	90.9	90.9	78.5	91.7	84.6	77.3	77.4	77.4
	Spanish	88.4	89.0	88.7	71.6	91.2	80.2	82.1	74.1	77.9
	French	88.0	92.4	90.1	68.0	91.0	77.8	85.2	74.9	79.7
	German	90.1	93.7	**91.9**	81.6	91.7	**86.4**	84.2	82.0	**83.0**
	Russian	84.4	88.2	86.2	51.4	91.1	65.7	73.6	77.2	75.4
	Arabic	62.4	93.8	74.8	53.3	89.7	66.8	81.8	48.1	60.6
Arabic	English	67.1	79.5	72.8	87.5	66.0	75.3	35.4	37.6	**36.5**
	Spanish	71.1	72.9	72.0	80.0	73.3	76.5	29.5	22.7	25.6
	French	68.0	73.4	70.6	80.5	72.8	76.4	39.8	23.9	29.8
	German	69.2	81.5	74.8	86.7	69.2	**77.0**	36.5	18.7	24.7
	Russian	67.7	76.6	71.8	77.3	75.6	76.4	14.6	6.3	8.8
	Arabic	66.6	87.3	**75.6**	84.0	70.0	76.3	30.4	26.7	28.4
Basque	English	62.3	80.8	**70.3**	60.8	70.8	**65.4**	40.8	18.9	**25.8**
	Spanish	60.5	74.6	66.8	47.3	79.5	59.3	41.4	14.0	20.9
	French	63.2	76.0	69.0	49.2	77.6	60.2	28.4	16.7	21.0
	German	60.7	78.6	68.5	57.8	74.8	65.2	47.3	11.0	17.8
	Russian	56.9	74.2	64.4	40.4	85.1	54.8	2.3	0.1	0.2
	Arabic	48.6	81.7	61.0	44.2	78.1	56.4	19.7	0.9	1.7
Bulgarian	English	88.6	95.7	92.0	86.7	86.3	86.5	80.0	55.0	65.2
	Spanish	86.0	95.6	90.5	84.2	87.6	85.9	80.2	47.3	59.4
	French	83.9	95.7	89.4	82.8	92.9	**87.6**	80.7	44.2	57.1
	German	86.6	95.7	90.9	86.5	84.6	85.6	78.2	54.4	64.2
	Russian	92.9	96.3	**94.6**	75.2	94.1	83.6	86.3	74.3	**79.9**
	Arabic	69.3	95.9	80.4	68.9	87.0	76.9	69.5	25.2	36.9

Finnish	English	81.1	89.2	**85.0**	80.1	81.1	80.6	68.5	59.9	**63.9**
	Spanish	80.0	85.6	82.7	67.7	84.8	75.3	67.1	50.5	57.6
	French	79.7	84.9	82.2	65.6	86.4	74.6	67.2	45.5	54.2
	German	80.1	88.7	84.2	82.1	79.9	**81.0**	65.6	57.6	61.3
	Russian	80.5	86.6	83.4	59.7	84.8	70.1	65.7	57.7	61.4
	Arabic	62.1	92.7	74.4	64.4	75.4	69.4	66.7	22.2	33.3
Georgian	English	80.9	86.8	83.7	73.8	95.8	83.4	87.5	61.7	72.4
	Spanish	79.5	82.8	81.1	67.4	95.5	79.0	83.2	64.3	72.5
	French	77.9	86.0	81.8	66.0	97.2	78.6	80.4	56.8	66.6
	German	78.7	89.7	**83.9**	76.8	94.1	**84.6**	86.7	64.6	**74.0**
	Russian	78.7	88.3	83.2	72.9	97.6	83.4	82.6	61.5	70.5
	Arabic	56.6	94.7	70.8	61.6	92.7	74.0	82.8	21.1	33.6
Hindi	English	69.6	86.0	76.9	55.4	70.1	61.9	55.8	53.3	54.5
	Spanish	67.2	88.7	76.5	50.6	82.2	62.7	55.9	47.2	51.2
	French	68.3	88.3	**77.0**	57.9	80.3	**67.3**	58.7	48.0	52.8
	German	68.7	84.7	75.8	53.5	76.7	63.1	60.4	51.6	**55.6**
	Russian	66.7	82.7	73.9	36.2	92.6	52.0	59.1	48.1	53.0
	Arabic	51.4	89.1	65.2	48.7	73.7	58.6	64.2	33.0	43.5
Indonesian	English	77.9	90.4	**83.7**	89.3	84.9	**87.0**	60.4	56.4	**58.3**
	Spanish	77.6	87.8	82.4	86.3	81.9	84.1	61.1	52.3	56.4
	French	72.2	90.3	80.3	83.0	89.5	86.1	61.8	39.7	48.3
	German	73.5	88.2	80.2	88.8	83.0	85.8	60.4	50.7	55.1
	Russian	74.1	90.5	81.5	77.1	90.3	83.2	55.5	37.9	45.0
	Arabic	59.6	92.7	72.5	80.0	82.2	81.1	50.8	40.9	45.3
Kazakh	English	70.1	85.3	**77.0**	61.0	82.7	**70.2**	62.8	19.2	**29.4**
	Spanish	69.3	60.5	64.6	39.3	93.4	55.3	69.4	3.3	6.3
	French	66.0	73.5	69.5	47.0	90.2	61.8	55.7	6.6	11.8
	German	65.9	79.1	71.9	51.3	88.0	64.8	0.0	0.0	0.0
	Russian	68.9	56.2	61.9	36.2	94.2	52.3	0.0	0.0	0.0
	Arabic	53.4	87.9	66.5	50.1	74.1	59.7	0.0	0.0	0.0
Lithuanian	English	83.1	93.4	88.0	89.6	79.5	84.2	53.0	56.5	54.7
	Spanish	81.2	92.4	86.5	81.8	85.8	83.7	55.4	42.9	48.3
	French	80.9	92.9	86.5	82.3	89.3	85.7	53.6	43.0	47.7
	German	81.3	93.4	87.0	88.3	78.1	82.9	54.1	58.2	56.1
	Russian	86.5	91.0	**88.7**	83.0	89.8	**86.3**	56.7	60.7	**58.6**
	Arabic	61.4	95.3	74.7	82.3	75.5	78.8	38.5	9.6	15.4

Persian	English	88.1	82.7	85.3	36.5	59.2	45.2	81.8	45.4	58.4
	Spanish	88.9	83.7	**86.2**	43.6	76.4	55.5	82.7	47.7	**60.5**
	French	87.5	84.5	86.0	40.3	72.1	51.7	81.5	38.1	51.8
	German	86.4	82.9	84.6	41.8	66.2	51.2	83.6	37.0	51.3
	Russian	89.8	80.5	84.9	39.9	93.0	55.9	73.2	51.1	60.2
	Arabic	81.1	89.0	84.9	43.2	80.7	**56.2**	83.7	38.1	52.4
Portuguese	English	88.7	92.1	90.4	84.9	89.8	87.3	66.1	66.5	66.3
	Spanish	91.1	94.9	**93.0**	90.3	86.0	88.1	73.6	83.6	**78.3**
	French	87.0	94.9	90.8	85.7	93.8	**89.6**	74.9	67.8	71.2
	German	82.4	94.2	87.9	86.6	88.9	87.7	67.8	46.7	55.3
	Russian	84.9	92.3	88.4	76.4	95.5	84.9	72.4	67.8	70.0
	Arabic	66.1	96.6	78.5	74.6	86.2	80.0	59.6	32.3	41.9
Telugu	English	79.9	63.2	**70.5**	80.9	90.9	**85.6**	25.3	40.0	**30.9**
	Spanish	73.9	59.8	66.1	65.7	95.7	77.9	44.4	20.0	27.4
	French	74.3	61.8	67.4	68.6	89.6	77.7	11.1	6.7	8.3
	German	71.7	56.1	63.0	75.3	92.8	83.1	16.7	6.7	9.5
	Russian	70.4	58.9	64.1	68.9	94.4	79.7	66.7	13.3	22.2
	Arabic	62.4	63.2	62.8	59.2	90.7	71.6	0.0	0.0	0.0
Turkish	English	75.5	80.7	**78.0**	80.4	79.6	80.0	72.9	35.2	**47.5**
	Spanish	75.5	76.0	75.7	72.0	86.0	78.3	75.1	27.8	40.6
	French	74.4	80.4	77.3	77.5	88.8	**82.8**	78.1	29.3	42.6
	German	74.6	77.7	76.1	78.4	86.9	82.4	73.7	28.0	40.6
	Russian	69.1	73.6	71.3	68.5	89.7	77.7	55.9	15.3	23.6
	Arabic	56.1	85.8	67.9	76.3	79.8	78.0	72.2	10.4	18.2

Table 2.1: The precision, recall and F1-score for nouns, verbs and adjectives per language pair when using the Bible as the source of parallel data. The best F1-score per target language and POS tag is in **bold**.

Tables 2.2 and 2.3 report the POS accuracy per target-source language pair and the average performance per source and target language using our word-based unsupervised cross-lingual POS-tagging system in the *No_MONO* and *No_XLM* ablation setups, respectively, when using the Bible as the source of parallel data and evaluating on the test sets of UD-v2.5. We use only the New Testament in the cases of Basque, Georgian and Kazakh, and we use in-house annotations for Georgian.

Target Language	Source for Unsupervised Learning						
	English	Spanish	French	German	Russian	Arabic	Average
Afrikaans	**83.5**	79.4	80.3	80.7	73.6	64.2	77.0
Amharic	**70.8**	**70.8**	69.8	70.5	65.1	68.8	69.3
Basque	**61.2**	57.8	57.8	**61.2**	57.4	49.2	57.4
Bulgarian	**81.7**	79.2	79.6	76.5	81.2	66.9	77.5
Finnish	**76.6**	73.6	74.0	75.4	71.0	59.6	71.7
Georgian	**75.6**	73.6	71.7	75.0	73.3	64.7	72.3
Hindi	**68.1**	65.3	65.4	54.6	59.7	49.8	60.5
Indonesian	**80.5**	79.5	79.1	77.2	75.0	65.8	76.2
Kazakh	**67.9**	61.3	62.2	64.9	58.5	59.2	62.3
Lithuanian	77.3	74.3	75.0	74.9	**78.8**	65.4	74.3
Persian	74.2	73.8	71.6	**74.7**	72.3	68.1	72.5
Portuguese	83.2	**85.9**	84.0	78.7	75.4	64.5	78.6
Telugu	**75.0**	68.4	69.1	73.0	67.3	59.4	68.7
Turkish	69.0	67.0	**69.2**	66.2	64.7	59.8	66.0
Average	**74.6**	72.1	72.1	71.7	69.5	61.8	70.3

Table 2.2: The average POS-tagging performance (accuracy) in the *No_MONO* ablation setup when using the Bible as the source of parallel data. The best results per target and per source language are in **bold**.

Target Language	Source for Unsupervised Learning						
	English	Spanish	French	German	Russian	Arabic	Average
Afrikaans	**86.4**	82.2	83.3	83.6	77.4	67.8	80.1
Amharic	71.4	71.4	71.0	71.1	68.3	**72.1**	70.9
Basque	**64.0**	61.7	61.5	63.5	59.9	52.5	60.5
Bulgarian	84.5	82.1	82.0	79.0	**84.9**	70.0	80.4
Finnish	**81.0**	78.1	78.1	79.8	75.7	64.5	76.2
Georgian	80.8	76.9	76.9	**80.9**	80.8	69.9	77.7
Hindi	**73.2**	70.2	71.7	61.1	65.3	54.0	65.9
Indonesian	**82.2**	82.1	81.1	79.3	78.8	68.7	78.7
Kazakh	**70.4**	61.0	64.9	67.2	63.3	60.4	64.5
Lithuanian	80.4	77.7	78.1	78.0	**82.4**	68.2	77.5
Persian	74.7	74.7	73.0	75.3	**75.9**	68.0	73.6
Portuguese	85.5	**88.0**	86.4	80.7	78.0	67.6	81.0
Telugu	**75.6**	67.5	71.3	72.5	69.6	62.1	69.8
Turkish	72.0	70.6	**72.5**	70.4	69.4	64.9	70.0
Average	**77.3**	74.6	75.1	74.5	73.5	65.0	73.3

Table 2.3: The average POS-tagging performance (accuracy) in the *No_XLM* ablation setup when using the Bible as the source of parallel data. The best results per target and per source language are in **bold**.

Table 2.4 reports the precision, recall and F1-score for nouns, verbs and adjectives per target-source language pair using our single-source word-based and stem-based unsupervised cross-lingual POS-tagging systems when using the New Testament as the source of parallel data and evaluating on the test sets of UD-v2.5. We use in-house annotations for Georgian.

Target	Source	Noun			Verb			Adjective			
Language	Language	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1	
Amharic	English	Word-Based	64.4	**86.6**	73.8	**84.1**	70.9	77.0	**49.4**	28.4	36.1
		Stem-Based	**72.5**	85.6	**78.5**	84.0	**81.0**	**82.5**	48.7	**34.5**	**40.4**
	Spanish	Word-Based	68.4	78.0	72.9	73.2	**79.2**	76.0	9.4	0.9	1.6
		Stem-Based	**69.2**	**87.8**	**77.4**	**83.9**	74.4	**78.9**	**43.0**	**21.3**	**28.4**
	French	Word-Based	**68.9**	79.5	73.8	74.1	**81.4**	77.5	**50.0**	0.6	1.1
		Stem-Based	68.6	**89.2**	**77.5**	**84.5**	75.5	**79.7**	49.5	**22.4**	**30.6**
	German	Word-Based	69.0	83.0	75.4	78.6	**80.8**	79.7	0.0	0.0	0.0
		Stem-Based	**69.1**	**89.5**	**78.0**	**85.1**	76.3	**80.4**	**73.2**	**20.4**	**31.9**
	Russian	Word-Based	69.9	72.0	70.9	64.0	**89.1**	74.5	0.0	0.0	0.0
		Stem-Based	**70.8**	**87.8**	**78.4**	**77.1**	80.6	**78.8**	**40.6**	**22.7**	**29.1**
	Arabic	Word-Based	62.8	**90.0**	74.0	77.0	72.5	74.6	17.0	6.9	9.8
		Stem-Based	**67.4**	89.8	**77.0**	**82.9**	**73.8**	**78.0**	**35.6**	**30.7**	**33.0**
Basque	English	Word-Based	62.3	80.8	70.3	60.8	**70.8**	65.4	40.8	18.9	25.8
		Stem-Based	**63.9**	**81.7**	**71.7**	**76.3**	67.8	**71.8**	**41.3**	**24.9**	**31.0**
	Spanish	Word-Based	60.5	74.6	66.8	47.3	**79.5**	59.3	**41.4**	14.0	20.9
		Stem-Based	**60.8**	**83.9**	**70.5**	**75.8**	67.3	**71.3**	36.0	**24.6**	**29.1**
	French	Word-Based	**63.2**	76.0	69.0	49.2	**77.6**	60.2	28.4	16.7	21.0
		Stem-Based	62.2	**84.3**	**71.6**	**73.6**	68.7	**71.1**	**34.1**	**25.4**	**29.0**
	German	Word-Based	60.7	78.6	68.5	57.8	**74.8**	65.2	**47.3**	11.0	17.8
		Stem-Based	**60.8**	**81.9**	**69.8**	**76.8**	66.8	**71.5**	43.1	**17.3**	**24.7**
	Russian	Word-Based	56.9	74.2	64.4	40.4	**85.1**	54.8	2.3	0.1	0.2
		Stem-Based	**58.9**	**83.6**	**69.1**	**54.9**	76.7	**64.0**	**33.0**	**19.5**	**24.4**
	Arabic	Word-Based	48.6	81.7	61.0	44.2	**78.1**	56.4	19.7	0.9	1.7
		Stem-Based	**51.6**	**89.1**	**65.4**	**66.4**	66.5	**66.4**	**32.1**	**19.7**	**24.4**

Finnish	English	Word-Based	76.0	**88.0**	81.6	**77.7**	80.8	79.2	**69.2**	44.7	54.3
		Stem-Based	**80.2**	87.0	**83.4**	77.0	**83.8**	**80.3**	62.8	**54.1**	**58.1**
	Spanish	Word-Based	**75.5**	83.5	79.3	63.0	**85.9**	72.7	**74.6**	31.6	44.3
		Stem-Based	75.0	**89.4**	**81.6**	**80.4**	78.8	**79.6**	59.9	**40.7**	**48.3**
	French	Word-Based	**76.4**	80.8	78.6	58.4	**88.3**	70.3	**71.9**	30.7	43.0
		Stem-Based	75.3	**89.4**	**81.8**	**75.6**	82.5	**78.9**	64.7	**43.3**	**51.9**
	German	Word-Based	73.7	88.3	80.4	79.8	81.1	80.4	**73.4**	31.3	43.9
		Stem-Based	**76.9**	**89.7**	**82.8**	**82.7**	**81.6**	**82.1**	69.6	**45.9**	**55.3**
	Russian	Word-Based	75.8	88.2	81.5	60.1	**85.2**	70.5	**69.1**	47.5	56.3
		Stem-Based	**78.8**	**88.4**	**83.3**	**62.7**	84.6	**72.0**	66.5	**54.2**	**59.7**
	Arabic	Word-Based	55.1	**93.8**	69.4	68.6	74.0	71.2	**63.5**	10.1	17.4
		Stem-Based	**61.9**	93.7	**74.5**	**76.6**	**78.0**	**77.3**	61.4	**29.7**	**39.9**
Georgian	English	Word-Based	**80.9**	**86.8**	**83.7**	73.8	**95.8**	83.4	87.5	**61.7**	**72.4**
		Stem-Based	79.6	86.0	82.7	**74.7**	**95.8**	**84.0**	**90.5**	56.0	69.1
	Spanish	Word-Based	**79.5**	82.8	81.1	67.4	**95.5**	79.0	83.2	**64.3**	**72.5**
		Stem-Based	76.6	**89.1**	**82.4**	**77.3**	92.7	**84.3**	**89.2**	58.1	70.3
	French	Word-Based	**77.9**	86.0	81.8	66.0	**97.2**	78.6	80.4	56.8	66.6
		Stem-Based	76.4	**89.5**	**82.5**	**77.3**	96.2	**85.7**	**82.3**	**58.1**	**68.1**
	German	Word-Based	78.7	**89.7**	83.9	**76.8**	94.1	84.6	86.7	**64.6**	**74.0**
		Stem-Based	**79.6**	89.0	**84.0**	76.6	**94.9**	**84.8**	**89.4**	61.2	72.6
	Russian	Word-Based	78.7	88.3	83.2	72.9	**97.6**	83.4	82.6	61.5	70.5
		Stem-Based	**80.7**	**88.5**	**84.4**	**76.1**	97.2	**85.4**	**82.7**	**63.0**	**71.5**
	Arabic	Word-Based	56.6	**94.7**	70.8	61.6	92.7	74.0	**82.8**	21.1	33.6
		Stem-Based	**67.4**	93.1	**78.2**	**83.4**	**93.2**	**88.0**	78.6	**51.8**	**62.4**
Indonesian	English	Word-Based	**76.1**	**90.6**	**82.7**	**88.9**	85.5	**87.2**	53.2	**44.0**	48.1
		Stem-Based	75.7	89.8	82.1	87.0	**86.1**	86.5	**54.3**	**44.0**	**48.6**
	Spanish	Word-Based	**73.3**	89.0	**80.4**	82.6	**88.5**	**85.5**	49.3	31.6	38.5
		Stem-Based	71.6	**90.2**	79.8	**86.9**	82.9	84.9	**55.1**	**33.0**	**41.3**
	French	Word-Based	**68.0**	89.6	**77.3**	77.4	**90.7**	83.5	**60.4**	29.6	**39.7**
		Stem-Based	66.5	**91.5**	77.0	**83.3**	85.8	**84.5**	56.4	**29.7**	38.8
	German	Word-Based	65.5	90.3	75.9	87.5	**82.6**	**84.9**	50.5	17.3	25.8
		Stem-Based	**66.4**	**90.6**	**76.6**	**88.4**	79.2	83.5	**71.9**	**36.5**	**48.5**
	Russian	Word-Based	67.7	88.8	76.8	68.2	**92.6**	78.6	27.0	15.3	19.1
		Stem-Based	**71.7**	**91.1**	**80.3**	**76.1**	86.9	**81.1**	**66.4**	**41.0**	**50.7**
	Arabic	Word-Based	52.6	93.6	67.4	74.3	**84.0**	78.8	47.9	18.8	27.0
		Stem-Based	**56.9**	**94.1**	**70.9**	**83.7**	76.5	**79.9**	**52.5**	**34.2**	**41.4**

Kazakh	English	Word-Based	70.1	85.3	77.0	61.0	**82.7**	70.2	62.8	19.2	29.4
		Stem-Based	**74.1**	**87.1**	**80.1**	**71.9**	79.9	**75.7**	**66.9**	**24.5**	**35.8**
	Spanish	Word-Based	69.3	60.5	64.6	39.3	**93.4**	55.3	**69.4**	**3.3**	**6.3**
		Stem-Based	**71.6**	**88.0**	**78.9**	**66.0**	79.8	**72.2**	67.9	19.3	30.1
	French	Word-Based	66.0	73.5	69.5	47.0	**90.2**	61.8	55.7	6.6	11.8
		Stem-Based	**69.4**	**90.8**	**78.7**	**68.6**	82.5	**74.9**	**69.9**	**23.5**	**35.2**
	German	Word-Based	65.9	79.1	71.9	51.3	**88.0**	64.8	0.0	0.0	0.0
		Stem-Based	**69.7**	**89.6**	**78.4**	**68.2**	77.9	**72.7**	**71.8**	**17.6**	**28.3**
	Russian	Word-Based	68.9	56.2	61.9	36.2	**94.2**	52.3	0.0	0.0	0.0
		Stem-Based	**71.4**	**84.5**	**77.4**	**61.5**	87.1	**72.1**	**68.8**	**17.0**	**27.2**
	Arabic	Word-Based	53.4	87.9	66.5	50.1	74.1	59.7	0.0	0.0	0.0
		Stem-Based	**61.6**	**91.8**	**73.7**	**68.1**	**76.5**	**72.0**	**60.9**	**11.2**	**18.8**
Telugu	English	Word-Based	**70.1**	61.0	65.2	68.7	**94.8**	79.7	**11.1**	**13.3**	**12.1**
		Stem-Based	69.4	**64.3**	**66.7**	**75.3**	91.7	**82.7**	4.2	6.7	5.1
	Spanish	Word-Based	**68.4**	47.4	56.0	54.0	**98.6**	69.8	0.0	0.0	0.0
		Stem-Based	65.7	**57.9**	**61.5**	**64.5**	95.0	**76.8**	0.0	0.0	0.0
	French	Word-Based	63.5	46.8	53.8	53.9	**99.2**	69.8	0.0	0.0	0.0
		Stem-Based	**67.4**	**61.6**	**64.4**	**62.4**	91.5	**74.2**	0.0	0.0	0.0
	German	Word-Based	**68.3**	50.9	58.3	55.8	93.4	69.9	0.0	0.0	0.0
		Stem-Based	68.2	**53.0**	**59.6**	**60.7**	**95.0**	**74.0**	0.0	0.0	0.0
	Russian	Word-Based	50.0	31.6	38.7	47.5	**98.8**	64.2	0.0	0.0	0.0
		Stem-Based	**61.9**	**48.9**	**54.5**	**58.3**	97.3	**72.9**	0.0	0.0	0.0
	Arabic	Word-Based	46.7	45.2	45.9	48.3	**96.9**	64.5	0.0	0.0	0.0
		Stem-Based	**51.4**	**74.3**	**60.7**	**74.3**	88.0	**80.5**	0.0	0.0	0.0
Turkish	English	Word-Based	75.4	78.5	76.9	74.7	**84.3**	79.2	**71.5**	**32.0**	**44.2**
		Stem-Based	**76.6**	**81.7**	**79.1**	**80.9**	80.9	**80.9**	68.9	29.0	40.8
	Spanish	Word-Based	**75.4**	68.1	71.5	61.9	**92.3**	74.1	**79.9**	20.1	32.1
		Stem-Based	75.3	**82.9**	**78.9**	**82.3**	76.9	**79.4**	71.2	**26.8**	**38.9**
	French	Word-Based	**75.2**	66.5	70.6	60.1	**93.5**	73.1	80.8	26.0	39.4
		Stem-Based	74.7	**85.5**	**79.7**	**82.7**	75.7	**79.0**	**81.6**	**28.2**	**41.9**
	German	Word-Based	69.0	76.4	72.5	68.1	**88.1**	76.8	79.4	10.2	18.0
		Stem-Based	**73.8**	**83.1**	**78.2**	**79.8**	79.7	**79.7**	**82.5**	**25.5**	**39.0**
	Russian	Word-Based	62.5	67.3	64.8	53.5	**94.0**	68.2	53.6	1.2	2.3
		Stem-Based	**74.7**	**80.9**	**77.7**	**76.5**	88.5	**82.1**	**77.7**	**28.9**	**42.1**
	Arabic	Word-Based	52.5	85.2	65.0	68.2	**82.8**	74.8	38.6	0.4	0.8
		Stem-Based	**66.6**	**87.3**	**75.6**	**82.2**	79.7	**80.9**	**57.6**	**20.9**	**30.7**

Table 2.4: The precision, recall and F1-score for nouns, verbs and adjectives per language pair in the single-source word-based and stem-based approaches when using the New Testament as the source of parallel data. The best F1-score per language pair for each evaluation metric is in **bold**.

Tables 2.5 and 2.6 report the POS accuracy when using the stems and affixes as learning features in our stem-based unsupervised cross-lingual POS-tagging system in the single-source and multi-source setups, respectively, when using the New Testament as the source of parallel data and evaluating on the test sets of UD-v2.5. We use in-house annotations for Georgian.

Target Language	Segmentation Features	Source for Unsupervised Learning					
		English	Spanish	French	German	Russian	Arabic
Amharic	None	79.6	77.5	77.7	**77.8**	76.2	74.5
	Stem	**80.2**	77.5	**78.0**	77.6	**76.6**	74.6
	Stem+Affixes	79.8	**77.7**	77.8	**77.8**	76.5	**74.7**
Basque	None	**69.1**	70.4	70.5	**69.6**	65.2	60.8
	Stem	68.7	70.5	70.5	69.3	**65.6**	60.3
	Stem+Affixes	69.0	**70.6**	**70.8**	69.1	65.3	**62.0**
Finnish	None	**81.9**	80.1	80.9	82.3	79.0	70.3
	Stem	**81.9**	**80.4**	80.9	**82.4**	**79.1**	70.5
	Stem+Affixes	81.8	80.1	**81.2**	**82.4**	78.9	**70.6**
Georgian	None	82.0	80.4	81.0	**82.2**	**83.4**	**79.0**
	Stem	**82.1**	**80.5**	**81.3**	82.1	83.3	78.7
	Stem+Affixes	81.5	80.3	80.9	81.7	83.1	78.8
Indonesian	None	**82.5**	**81.0**	**80.1**	77.3	81.2	**72.3**
	Stem	**82.5**	80.8	79.9	**77.6**	**81.3**	71.7
	Stem+Affixes	**82.5**	80.9	80.0	77.3	81.0	72.0
Kazakh	None	76.4	74.8	75.5	**73.2**	**73.6**	**70.8**
	Stem	76.3	74.8	75.7	72.8	**73.6**	70.7
	Stem+Affixes	**76.6**	**75.2**	**75.8**	73.1	**73.6**	**70.8**
Telugu	None	**78.6**	**72.7**	72.2	**71.9**	**69.6**	66.8
	Stem	77.9	71.5	**72.7**	**71.9**	**69.6**	66.7
	Stem+Affixes	78.4	72.4	**72.7**	71.4	68.7	**67.1**
Turkish	None	**73.7**	**73.1**	73.0	**71.9**	**77.6**	**71.9**
	Stem	73.5	73.0	**73.1**	71.5	**77.6**	71.7
	Stem+Affixes	73.6	73.0	**73.1**	71.8	**77.6**	71.7

Table 2.5: The POS-tagging performance (accuracy) of the single-source stem-based setup with the use of different segmentation features when using the New Testament as the source of parallel data. The best result per target-source language pair is in **bold**. The improvements that are due to the use of segmentation features that are statistically significant for $p\text{-}value < 0.01$ are underlined.

Target Language	Segmentation Features	Multi-source Setup							
		MP_{wmv}	MP_{bys}	MP_{wbys}	MD_{wmv_a}	MD_{wmv_d}	MD_{bys}	MD_{wbys_a}	MD_{wbys_d}
Amharic	None	79.6	80.4	**80.8**	78.6	79.1	79.2	79.2	79.2
	Stem	**79.7**	**80.8**	80.7	**78.7**	**79.2**	**79.5**	**79.5**	**79.7**
	Stem+Affixes	79.4	80.2	80.4	**78.7**	79.1	79.2	79.3	79.3
Basque	None	71.4	71.7	71.7	**71.0**	72.0	71.9	72.3	71.8
	Stem	71.6	71.9	71.9	70.9	**72.1**	**72.6**	**72.6**	**72.6**
	Stem+Affixes	**71.8**	**72.1**	**72.0**	70.9	72.0	71.8	71.8	71.6
Finnish	None	**82.9**	**82.7**	**82.5**	82.4	82.7	**83.2**	83.0	83.0
	Stem	82.7	82.6	82.2	**82.7**	82.8	83.1	83.0	83.0
	Stem+Affixes	**82.9**	**82.7**	82.4	**82.7**	**82.9**	**83.2**	**83.2**	**83.2**
Georgian	None	84.7	**84.5**	**84.2**	84.3	84.3	**84.5**	**84.4**	**84.5**
	Stem	**85.0**	84.1	83.9	**84.4**	**84.5**	84.3	84.2	84.3
	Stem+Affixes	84.3	84.2	**84.2**	83.7	83.8	84.2	84.1	84.1
Indonesian	None	**81.0**	81.0	**80.9**	**81.4**	**81.9**	**82.0**	82.0	82.0
	Stem	80.9	80.8	**80.9**	81.1	81.5	81.8	81.8	81.9
	Stem+Affixes	80.6	**81.1**	**80.9**	81.0	81.6	**82.0**	**82.1**	**82.1**
Kazakh	None	76.7	76.8	76.9	75.3	75.8	76.7	76.6	76.6
	Stem	76.5	77.0	76.8	**75.4**	**76.0**	76.8	76.7	76.7
	Stem+Affixes	**76.8**	**77.3**	**77.0**	75.3	**76.0**	**76.9**	**76.8**	**76.8**
Telugu	None	**73.8**	71.7	**70.8**	72.9	73.6	**73.4**	73.4	73.4
	Stem	73.1	71.7	**70.8**	73.1	73.8	73.3	**73.6**	**73.6**
	Stem+Affixes	73.7	**72.1**	70.7	**73.6**	**74.0**	73.3	**73.6**	73.5
Turkish	None	73.6	73.6	73.7	**75.4**	**75.2**	**74.4**	**74.4**	74.3
	Stem	73.7	73.7	73.7	75.1	75.1	74.1	74.3	74.1
	Stem+Affixes	**73.9**	**73.8**	**74.0**	75.3	75.1	74.3	**74.4**	**74.4**

Table 2.6: The POS-tagging performance (accuracy) of the multi-source stem-based setups with the use of different segmentation features when using the New Testament as the source of parallel data. The best result per {target and multi-source setup} pair is in **bold**. The improvements that are due to the use of segmentation features that are statistically significant for $p\text{-}value < 0.01$ are underlined.