Morphological Process Transduction:
Towards interpretable multi-lingual morphological analysis

Ronald Ahmed Cardenas Acosta

MSc. Dissertation

Department of Artificial Intelligence
Institute of Linguistics and Language Technology
Faculty of Information and Communication Technology

University of Malta

September, 2019

Supervisor(s):
Dr. Claudia Borg, Ph.D., Department of Artificial Intelligence, University of Malta
,
RNDr. Daniel Zeman, Ph.D., Institute of Formal and Applied Linguistics,
Charles University in Prague

Submitted in partial fulfilment of the requirements for the degree of
Master of Science in Human Language Science and Technology (HLST)

FACULTY OF
INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITY OF MALTA

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I, the undersigned, declare that the Masters dissertation submitted is my own work, except where acknowledged and referenced.

I understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Student Name:	Ronald Ahmed Cardenas Acosta
Course Code:	CSA5310 HLST Dissertation
Title of work:	Morphological Process Transduction:
Towards interpretable multi-lingual morphological analysis	

Signature of Student:

Date:

Supervisor(s)

Dr. Claudia Borg

Department of Artificial Intelligence

University of Malta

[and

RNDr. Daniel Zeman

Institute of Formal and Applied Linguistics,

Charles University in Prague]*

Local Advisor(s)

Dr. Claudia Borg

Department of Artificial Intelligence

University of Malta

Acknowledgements

To my parents, whose sacrifice and courage engraved into me the meaning of commitment, and whose continuous support kept me focused on the goal.

To the LCT organization, for giving the amazing opportunity of walking this path. I will be eternally grateful.

To my supervisors, Dr. Claudia Borg and Dr. Daniel Zeman, for the valuable guidance and feedback they always had for me.

And last but not least, to the family I made along my journey, precious friends in Prague and Malta.

Abstract

The persistent efforts to make valuable annotated corpora in more diverse, morphologically rich languages has driven research in NLP into considering more explicit techniques to incorporate morphological information into the pipeline. Recent efforts have proposed combined strategies to bring together the transducer paradigm and neural architectures, although ingesting one character at a time in a context-agnostic setup. In this thesis, we introduce a technique inspired by the *byte-pair-encoding* (BPE) compression algorithm in order to obtain transducing actions that resemble word formations more faithfully. Then, we propose a neural transducer architecture that operates over these transducing actions, ingesting one word token at a time and effectively incorporating sentence-level context by encoding per-token action representations in a hierarchical fashion. We investigate the benefit of this word formation representations for the tasks of lemmatization and context-aware morphological tagging for a typologically diverse set of languages.

For lemmatization, we use investigate an optimization technique that explores possible action sequences and scores them based on task-specific metrics instead of standard log-likelihood. We find that our approach benefits greatly languages that use less commonly studied morphological processes such as templatic processes, with up to 55.73% error reduction in lemmatization for Arabic. Furthermore, we find that projecting these word formation representations into a common multilingual space enables our models to group together action labels signaling the same phenomena in several languages, e.g. Plurality, irrespective of the language-specific morphological process that may be involved.

For morphological tagging, we investigate the effect of different tagging strategies, e.g. bundle vs individual tag prediction, as well as the effect of multilingual action representations. We find that our taggers are able to obtain up to 20% error reduction by leveraging multilingual actions with respect to the monolingual scenario.

Contents

Introduction

According to typological databases, the number of languages in the world ranges from 7111, as cataloged by Ethnologue [Eberhard et al., 2019], to 8494, as attested by Glottolog [Hammarstrm et al., 2019]. Yet, current research in NLP is limited to the languages for which linguistic annotations are available. In the last few years, impressive efforts have been made to consistently increase the number of covered languages. Examples of such efforts include the Universal Dependencies project [Nivre et al., 2019][1] and the UniMorph project [Kirov et al., 2018], featuring annotations for 146 and 111 languages, respectively. However, the annotation of such massive corpora is costly and time consuming. For this reason, many lines of research resort to unsupervised learning approaches in order to alleviate the necessity of annotated corpora. Even though recent lines of research feature unsupervised approaches to complex tasks such as Machine Translation [Lample et al., 2018a], the largest coverage reported to date is of 122 languages [Artetxe and Schwenk, 2018].

As the development of language technologies shifts to a more inclusive stance, the importance of explicitly modeling morphology becomes more evident. Recent efforts to include signals below the word level include encoding tokens character by character [Kim et al., 2016] or representing types with subword units [Sennrich et al., 2016, Kudo, 2018]. The methods to obtain these subword units, although unsupervised, are designed to capture regularities in contiguous spans of surface word forms such as the ones observed in polysynthetic or agglutinating languages. However, this approach fails to model regularities in non-contiguous spans such as the ones present in templatic languages. In addition, this approach does not model the underlying morphological mechanisms a language may be using in the process to go from lemma to final word form. These underlying mechanisms are known as *word formation processes*, and they will the focus of study in this work.

Word formation processes, oftentimes called morphological processes, are mechanisms by which a language modifies a lemma to accommodate a specific syntactic and semantic need in a sentence. In this thesis, we explore the idea of defining word formation processes

[1] Last edition at time of writing is v2.4

Language	Lemma	Word Form	Processes Involved	Processes as actions
English	book	book**s**	suffixation	suffixate(s)
Czech	kniha	knih**y**	subtraction + suffixation	substract(a) + suffixate(y)
Arabic	kitab	**al**k**u**t**u**b	prefixation + transfixation	prefixate(al) + transfixate(k_t_b,_u_u_)

Table 1: Example of how languages combine different word formation processes during inflection to encode Plurality. Surface segments involved in the processes are showed in bold.

as common ground for modeling how languages combine different processes during word formation. Consider the example in Table 1. Here we can see how English, Czech, and Arabic –presented in latin script for convenience– inflect word forms to encode Plurality into the noun *book*. We observe that English uses only one word formation process (suffixation), Czech uses two (substraction and suffixation), and Arabic also uses two (prefixation and transfixation).

The explicit modeling of word production operations opens the possibility to capture other morphological processes besides affixation or substraction, e.g. transfixation, and how these operations can signal morphological phenomena, e.g. Plurality, in different languages. In this thesis we take a step in this direction by posing word formation processes as 'actions' that sequentially edit a word form. In our example in Table 1, actions encode what process to perform (e.g. *suffixate*) and the segment involved (e.g. *-s*). We propose edit actions that resemble morphological processes and investigate how they can benefit the tasks of context-aware lemmatization and morphological tagging.

On the one hand, the task of lemmatization consists of mapping an inflected word form to its lemma, i.e. its dictionary form. In Table 2, for example, the form *sang* is mapped onto *sing*. On the other hand, the task of morphological tagging consists of mapping an inflected word form onto its morphosyntactic description (MSD) label. In the example in Table 2, *sang* is mapped onto the label `V;PST;IND;FIN` to indicate that this word form is a finite verb in past tense and indicative mood. In this thesis, we tackle the context-aware variant of these tasks, which means that the input to the system is a complete sentence instead of a single word form.

Inflected word form	Lemma	Morphosyntactic Description (MSD)
Tim	Tim	N;SG
sang	sing	V;PST;IND;FIN
carols	carol	N;PL

Table 2: Example of context-aware lemmatization and morphological tagging.

Previous work has posed the tasks of lemmatization and reinflection (mapping a lemma to its inflected form) as a string transduction problem, traditionally tackled using weighted finite state transducers [Eisner, 2002, Mohri, 2004]. More recently, however, neural transducers have been proposed. These architectures transduce one character at a time by using a set of operations based on edit-distance actions [Makarov and Clematide, 2018c,a, Schröder et al., 2018].

Follow up work further explored a variety of training strategies besides maximum likelihood. Makarov and Clematide [2018c] investigated the effect of exploration-based refinement of edit-distance operations by minimizing the expectation of a metric-driven risk, obtaining promising results on low-resource scenarios. Later on, Makarov and Clematide [2018b] proposed an imitation learning procedure that further eliminates the requirement of gold edit-distance alignments between lemmas and inflected forms. It is worth noting, however, that all these architectures transduce one character at a time and have no access to sentence-level context, viz. they solve context-agnostic tasks. In addition, even though these architectures were tested in several languages, they were trained on a monolingual setup and do not leverage the potential benefit of defining a language-agnostic set of edit-distance actions. Previous work that does focus on multilingual training of neural transducers is limited to learning a joint vocabulary of subword units [Kondratyuk, 2019]. Besides the splendid progress made so far, no previous work at the time of writing this work has addressed the question of what kind of morphological phenomena these actions are learning.

In regards to morphological tagging, previous work has explored the following two strategies: (i) tagging the complete MSD label, also known as 'bundle' [Kondratyuk, 2019, Üstün et al., 2019], e.g. '`N;PL`', and (ii) tagging the fine-grained feature components individually [Bhat et al., 2019], e.g. as '`N`' and '`PL`'. Later on, Straka et al. [2019] proposed

to combine both tagging strategies by learning to predict both schemes under a multi-task setup. These systems operate over subword units instead of edit-distance actions and once again, it is not clear what kind of morphological phenomena is being individually captured by these units.

In summary, the contributions of this thesis are the following:

- We introduce a technique based on the *byte-pair-encoding* (BPE) algorithm that produces edit actions that resemble morphological processes more faithfully. These actions operate at the word level instead of consuming one character at a time as in previous work [Makarov and Clematide, 2018c, Aharoni and Goldberg, 2016].
- We propose neural network architectures that leverage these action representations and incorporate sentence-level context in a hierarchical manner, for the tasks of lemmatization and morphological tagging in context.
- We provide a thorough analysis of exploration-based refinement of such representations under a reinforcement learning framework.
- We investigate the effect of multi-lingual projection of these action representations and how they can capture the same morphological phenomena in different languages, irrespective of the language-specific morphological processes involved.

Research Questions

We aim to shed light on the following research questions.

- What training strategies are more effective for learning edit operations resembling morphological processes?
- What kind of morphological phenomena can be captured by these edit actions? Can these actions learn to signal these phenomena in a multilingual setting?
- What morphological tagging strategy, e.g. bundle vs individual component prediction, is most benefited by morphological process representations?

Summary of Chapters

Chapter 02. Theoretical Background We begin by laying out the fundamental concepts and notation definitions that will be referred to throughout this thesis. The chapter spans a variety of topics, from morphology and its annotation schemes to optimization techniques in reinforcement learning.

Chapter 03. Literature Review In this chapter, we review the most relevant research work in morphological string transduction and how neural networks are being used for morphological analysis tasks.

Chapter 04. Transducing Pseudo Morphological Processes for Lemmatization and Morphological Analysis in Context In this chapter, we introduce an unsupervised method to obtain pseudo morphological operations, i.e. operations that resemble morphological processes and can be ingested by a transducer. We investigate the effectiveness of our method for the tasks of lemmatization and morphological tagging in context. We further explore multi-lingual projections and reinforcement learning as ways to transfer knowledge from more highly resourced languages.

Chapter 05. Experimental Setup In this chapter, we layout the details of our experiments including proposed models, evaluation metrics, and preliminary results on MRT tuning. In addition, we describe our participating system at the SIGMORPHON 2019 Shared Task.

Chapter 06. Results and Discussion We evaluate the performance of our models according to the metrics and perform error analysis experiments in order to shed light on what our models are learning. In addition, we talk about the limitations of our approach.

Chapter 07. Conclusions and Future Work First, we draw conclusions from the results presented and articulate on answers to the research questions presented in this introduction. Second, we comment on attractive future research paths that could be followed to tackle the main shortcomings of our approach.

1 Theoretical Background

In this chapter we lay out key concepts that will be referred to throughout this thesis. We start by defining what a morphological process is and what kinds of processes we are going to consider. Then, we comment on the most prominent current efforts in harmonization of linguistic annotations across languages. Later on, we elaborate on the original byte-pair-encoding algorithm and how it is applied to subword unit learning. Finally, we elaborate on the sub-field of Reinforcement Learning and its advantages over other learning paradigms, as well as the main optimization approaches used in our experiments.

1.1 Morphological Processes

A morphological process is the process by which a word form is transformed into another form by means of addition, subtraction or replacement of non-necessarily contiguous (and possibly empty) morphemes into its stem [Matthews, 1991]. These processes refine the encoded meaning and grammatical relations between the new word form and its context. A process is called *inflectional* when the grammatical category of the word form is not changed and the change in meaning, if any, results in a predictable, non-idiosyncratic drift. In contrast, a *derivational* process produces a greater idiosyncratic change of meaning but not necessarily changes the grammatical category. However, the line between derivational and inflectional morphology is sometimes blurry. For example, it results rather ambiguous to classify morpho-syntactic operations that have no overt realization, i.e. processes involving zero morphemes.

Morphological processes are classified into:

- **Affixation:** Addition of affix (suffix or prefix).
- **Circumfixation:** Addition of suffix and prefix.
- **Infixation:** the morpheme, infix, is inserted inside the stem.
- **Transfixation:** the transfix, a discontinuous affix, is inserted into a stem root or template.
- **Reduplication:** the whole stem or part of it is repeated.

- **Modification:** change in the phonetic substance of the stem. In this category we have vowel modification, vowel reversal, tonal and stress modification, consonant modification, and suppletion (replacement of one stem with another).

- **Subtraction:** Removal of a segment from the stem.

1.2 Morphological Processes Transduction

Oftentimes, a language will use more than one morphological process, one after the other, in order to encode a specific phenomena. Consider the case of transducing or transforming the arabic lemma *kitab* (book) into *alkutub* (books), an example depicted in Table 1. The lemma has to undergo through two morphological processes, prefixation and transfixation, in order to encode plurality.

We formalize the idea of applying a morphological process by reformulating these processes as 'actions' that transform a word form. In our example, the sequence of processes `prefixation, transfixation` is posed as `prefixate(al), transfixate(k_t_b,_u_u_)`. The first action indicates that the prefix `al` must be added, whereas the second action indicates that the transfix `_u_u_` must be added (or fused) to the root `k_t_b`.

We name the execution of actions that represent morphological processes as *morphological process transduction*.

1.3 Harmonization of linguistic annotations

Linguistic annotations are information added to raw language data in order to describe or analyze language under certain linguistic formalism. The structural complexity of linguistic annotations depends on the linguistic phenomena being described. For example, the description of the syntactic category of a word –aka Part-of-Speech (POS)– may require a single gloss, whereas the description of the dependency between words in a sentence is annotated as a tree –aka dependency tree.

The annotation process of new language data follows a scheme specially designed by expert linguists with the purpose of capturing linguistic phenomena of interest in the language being analyzed. Hence, the proposed glosses or structures are language specific.

In this context, the idea of harmonization of linguistic annotations emerged. Harmonization consists in mapping annotation sets designed for one language into a target annotation set. For example, we might want to map POS labels designed for Spanish into POS labels designed for English.

Early harmonization efforts targeted to create a target set common to the languages being analyzed. Projects such as EAGLE[2], PAROLE [3], and MULTEXT[4] aimed to standarize POS taget annotation among most European languages. However, the analysis of languages not covered in such projecs required tailored, often unidirectional, mapping between the source and target tagset. Later on, Zeman [2008] proposed a practical approach that minimized the effort incurred by annotators when analyzing non-covered languages. The approach drew ideas from the concept of "interlingua", an intermediate representation of meaning between languages. Theoretically, a translation system would map meanining from text in a source language into the interlingua, and then map this interlingua representation into the target language. Zeman [2008] proposed a language-agnostic tagset, *Interset*, meant as an intermediate mapping step for POS and morphology annotations. Then, a source–Interlingua mapper could be coupled with any Interlingua–target mapper. Subsequent efforts to define a language-agnostic POS tagset include early work from Petrov et al. [2012] and later on the Universal Dependencies (UD) project [Nivre et al., 2015], which now includes language-agnostic annotations of morphological features and dependency trees. More recently, the UniMorph project [Kirov et al., 2018] was proposed as an alternative universal morpho-syntactic annotation scheme. We now elaborate on the key features and differences of the UD and UniMorph conventions.

1.3.1 Universal Dependencies

With planned releases of new treebanks every six months, the Universal Dependencies project aims to provide linguistic resources with language-agnostic annotations for Part-of-Speech, morpho-syntactic features, and syntactic dependency relations. The latest release to the date of writing, v.2.4, features no less than 146 treebanks for 83 languages,

[2] `http://www.ilc.cnr.it/EAGLES96/home.html`
[3] `https://www.scss.tcd.ie/SLP/parole.htm`
[4] `https://cordis.europa.eu/project/rcn/19596/factsheet/en`

with 16 more treebanks awaiting to pass final sanity check tests.

UD proposes a coarse universal POS tagset with 17 tags. Additional lexical and grammatical properties can be encoded using what they call "universal features", an extensive tagset designed to account for linguistic phenomena at the morpho-syntactic level. The universal features' scheme is divided in two main categories, lexical and inflectional, further subdivided into a total of 49 subcategories. The key advantage of this annotation scheme is that it is designed to be extensible, i.e. subcategories and subcategory values can be added in order to accommodate a new phenomenon attested in a newly covered language. Another advantage of the UD scheme is that it is easily readable to non-expert users, hence expanding the target user audience.

1.3.2 UniMorph

The UniMorph project [Sylak-Glassman, 2016, Kirov et al., 2018] proposes a scheme specially designed to describe morphological features involved in inflectional morphology. The scheme defines 23 categories, defined as "dimensions of meaning", spanning a total of 212 category values. One such dimension is dedicated to POS categories. However, the POS tagset covers 8 categories and is based on the more functionally-motivated conceptual space proposed by Croft [2000]. In comparison with the UD scheme, this scheme is not extensible. However, it provides a comprehensive list of category values covering most morphological phenomena attested so far. Another difference w.r.t. the UD scheme is that the annotation scheme is designed to be compact and short, and hence, it is not easily readable for non-expert users.

1.4 Byte pair encoding and subword unit representation

Byte pair encoding (BPE, Gage [1994]) is a compression algorithm initially proposed to operate over a stream of bytes. The algorithm starts by finding the most frequent pair of adjacent bytes and replaces all instances of the pair by a single byte not seen in the stream. This process is repeated until no more unseen bytes are available or no more frequent pairs are found. One advantage of BPE with respect to other compression algorithms is that it never increases the size of the stream. This feature makes BPE especially suited for

applications with limited memory such as the representation of a string of characters, e.g. natural language text.

The encoding or representation of natural language text presents the following two extreme paradigms: (i) by means of a table of individual characters and (ii) by means of a table of distinct word forms, a.k.a. the vocabulary. A middle ground paradigm was proposed by Sennrich et al. [2016] by adapting the BPE algorithm to obtain a table of distinct contiguous character segments, namely *subword* units. The algorithm produces a table with less than or equal entries than a word form vocabulary would require. Moreover, the algorithm effectively takes advantage of regularities in inflected word forms such as common prefixes and suffixes.

The algorithm proposed by Sennrich et al. [2016] operates as follows. Given a stream of characters, the algorithm will iteratively merge the most frequent adjacent pair of segments (single characters in the beginning) for a pre-determined number of iterations. It is worth noting that merge operations take word boundaries into consideration, i.e. pairs that cross word boundaries are not merged. Hence, the algorithm can operate over a dictionary of word types weighted by their frequency. For example, given the dictionary { `'studied', 'played'`}, the first merge operation would be `('e','d')` $\mapsto$ `'ed'`.

1.5 Reinforcement Learning

Reinforcement Learning (RL) is a paradigm of learning that focuses on the interaction with an environment and observing how it reacts to a given set of actions. This paradigm introduces the concept of reward, a measurement of how effective an action is. The goal is to learn what action to perform next so that the reward is maximized.

The entity interacting with the environment is called *agent*, and it must learn which actions are most beneficial in the long run, i.e. it has to learn how and when to explore new actions based on what can be considered a vague concept of delayed reward in the case benefit cannot be immediately assessed.

Sutton and Barto [2018] formalized these characteristics in three aspects of the learning framework, namely sensation, action, and goal. First, *sensation* refers to the capacity of an agent to measure the environment of interest. The agent encodes this information as

a 'state'. Second, the *action* aspect refers to the capacity of the agent to act upon the environment. Lastly, the *goal* aspect refers to the learning goal of producing a sequence of actions that maximize a pre-defined reward.

In the last few years, RL has been increasingly applied to a wide range of NLP tasks in conjunction to underlying sequence2sequence (seq2seq) neural architectures, from morphological reinflection [Makarov and Clematide, 2018b] to machine translation [Shen et al., 2015] and summarization [Pasunuru and Bansal, 2018, Narayan et al., 2018].

1.5.1 Comparison of RL with other learning paradigms

Consider situations in which the feature space is dense, the sequence of actions to perform is long, or an environment is too complex to generalize over. It soon becomes unfeasible to have enough categorized samples that characterize correctly the task at hand. In contrast to supervised learning, reinforcement learning relies on the exploration of new ways of achieving better rewards and learning from its own mistakes while doing so. In contrast, reinforcement learning relies on the exploration of data points not attested in the training data. RL relies on the evaluation of a reward measure in order to decide whether an explored data point is meaningful. Hence, an agent learns from its own mistakes.

However, RL also presents some disadvantages, the most significant one being related to the specific exploration mechanism used and the complexity of the environment to be explored. Since a true model of the real environment is impossible to obtain, RL relies on the approximation of the environment by sampling information from it. Sampling requires to make –oftentimes strong– assumptions about the probability distribution we want to approximate, e.g. assuming that the feedback noise follows a Gaussian distribution. Since the true distribution is unknown, we are at risk of underestimating the environment, which in turn could make the model perform the wrong action.

1.5.2 Benefits of RL for sequence-to-sequence tasks

Tasks involving the mapping of one input sequence into an output sequence are known as sequence-to-sequence (or seq2seq) tasks. Previous work [Ranzato et al., 2015, Wiseman and Rush, 2016] has identified two main biases seq2seq models incur on during training: exposure bias and loss-evaluation mismatch. We now elaborate on what each of this

consist of and how reinforcement learning poses sensible solutions to these undesirable biases.

Exposure bias vs Exploration-exploitation Consider the case of language modelling. At training time, the model is only exposed to gold token sequences in order to learn the probability of the next work. However, at test time the model is expected to generate the next token based on its own previous prediction. This disparity between training and inference settings is referred to as *exposure bias*.

In this setting, a model cannot learn from its own mistakes because it is simply not exposed to them at training time. On the other hand, RL relies on a exploration-exploitation trade-off, i.e. an agent must learn to decide whether to explore new, less profitable actions or exploit actions that are known to contribute highly to the reward.

Loss-evaluation mismatch Another drawback of learning paradigms besides RL is the mismatch between the metric being optimized and the metric used for evaluation. Consider the case of machine translation trained to minimize the log likelihood of the data but it is evaluated using, for example, BLEU. A valid counter-argument, however, is that loss functions such as log likelihood and cross-entropy are differentiable, hence a variety of optimization algorithms can be applied.

In contrast, reward-driven training allows to optimize a model with respect to a evaluation metric. A loss function defined on this terms might end up being not differentiable. For this reason, RL training strategies rely on sampling to estimate complex optimization objectives.

One such training strategy is Minimum Risk Training (MRT). MRT tackles the previously mentioned training biases in a direct manner. First, MRT tackles exposure bias with exploration-exploitation trade-off over the target sequence. Second, MRT introduces evaluation metrics as part of the loss function and proceeds to optimizes the model parameters so as to minimize the expected loss on the training data. Previous work has employed MRT to optimize neural sequence-to-sequence architectures for the tasks of machine translation [Shen et al., 2015], and morphological reinflection and lemmatization [Rastogi et al., 2016, Makarov and Clematide, 2018c] with promising results.

1.5.3 Maximum Likelihood Estimate Optimization

Given a source sequence $x = \langle x_1, ..., x_n, ..., x_N \rangle$, and a target sequence $y = \langle y_1, ..., y_m, ..., y_M \rangle$, the aim is to train a model that consumes x and outputs y. Let us define the probability of sequence y as

$$P(y|x;\theta) = \Pi_{m=1}^{M} P(y_m|x, y_{<m};\theta) \tag{1}$$

where θ represents the model parameters and $y_{<m} = \langle y_1, ..., y_{m-1} \rangle$. Then, the model can be trained by maximizing the likelihood of training data $\mathcal{T} = \{\langle x^{(i)}, y^{(i)} \rangle\}_{i=1}^{|\mathcal{T}|}$, as follows

$$\hat{\theta}_{MLE} = \underset{\theta}{\text{argmax}}\{\mathcal{L}(\theta)\} \tag{2}$$

where $\mathcal{L}(\theta) = \sum_{i=1}^{|\mathcal{T}|} logP(y^{(i)}|x^{(i)};\theta)$. This optimization strategy is known as *Maximum Likelihood Estimate* (MLE) training, and it is known to suffer from exposure bias and loss-evaluation mismatch as pointed out by Ranzato et al. [2015], Wiseman and Rush [2016].

1.5.4 Minimum Risk Training

We now layout the concept of minimum risk and how to optimize it in the context of sequence to sequence prediction. Given training sample $\langle x^{(i)}, y^{(i)} \rangle$, let $\Delta(y, y^{(i)})$ be the loss function that quantifies the differences between the predicted sequence y and the gold sequence $y^{(i)}$. This loss function is not parameterized w.r.t. our model and hence, it is not differentiable. Then, the *risk* is defined as the expectation of the loss function w.r.t. the posterior distribution defined by Equation1. Hence, as introduced by Shen et al. [2015], the risk is defined by the expression

$$\mathcal{R}(\theta) = \sum_{i=1}^{\mathcal{T}} \mathbb{E}_{y|x^{(i)};\theta}\big[\Delta(y, y^{(i)})\big] \tag{3}$$

$$= \sum_{i=1}^{\mathcal{T}} \sum_{y \in \mathcal{Y}(x^{(i)})} P(y|x^{(i)};\theta)\Delta(y, y^{(i)}) \tag{4}$$

where $\mathcal{Y}(x^{(i)})$ is the set of all possible target sequences valid for source sequence $x^{(i)}$. Then, the objective is to minimize

$$\hat{\theta}_{MRT} = \underset{\theta}{\text{argmin}}\{\mathcal{R}(\theta)\} \tag{5}$$

Note that since $\Delta(y, y^{(i)})$ does not depend on θ, we do not need to differentiate it when calculating partial derivatives $\delta\mathcal{R}(\theta)/\delta\theta$. However, the search space $\mathcal{Y}(x^{(i)})$ in Equation 4 is oftentimes exponential, hence rendering the calculation of the expectations intractable. In this scenario, Shen et al. [2015] proposed to sample $\mathcal{Y}(x^{(i)})$ in order to approximate the posterior distribution $P(y|x^{(i)};\theta)$. Then, the optimization objective is defined as

$$\mathcal{R}(\theta) = \sum_{i=1}^{\mathcal{T}} \sum_{y \in \mathcal{S}(x^{(i)})} Q(y|x^{(i)};\theta)\Delta(y, y^{(i)}) \tag{6}$$

where $\mathcal{S}(x^{(i)}) \subset \mathcal{Y}(x^{(i)})$ is the subsampled space and $Q(y|x^{(i)};\theta)$ is the surrogate posterior defined by

$$Q(y|x^{(i)};\theta) = \frac{P(y|x^{(i)};\theta)^{\alpha}}{\sum_{\hat{y} \in \mathcal{S}(x^{(i)})} P(\hat{y}|x^{(i)};\theta)^{\alpha}} \tag{7}$$

with hyper-parameter α controlling the sharpness of posterior Q.

2 Literature Review

In this chapter we review relevant lines of research related to sequence transduction, focusing on string transduction. We name the transduction between a lemma and an inflected form (or vice versa) 'morphological string transduction'. We then survey how neural approaches have been implemented for morphological string transduction and tagging for low resource scenarios.

2.1 Neural Transducers

Many NLP tasks can be posited as the problem of transforming or *transducing* a sequence of information packages, e.g. words in one language, into another sequence that encodes information relevant to the task, e.g. words in another language. Tasks like these include machine translation, summarization, and speech recognition, to name a few. Before the advent of neural networks in the last few years, however, transducing systems used to resort to segmentation heuristics, hand-crafted features, and alignment models. In the case of morphological string transduction tasks such as reinflection or lemmatization, the traditional way to tackle these problems was with weighted finite state transducers (WFST, Mohri [2004], Eisner [2002]).

Early efforts in sequence transduction using neural networks modeled all possible alignments between the input and output sequence and its importance for phoneme recognition [Graves, 2012]. The idea of a fully differentiable alignment module was later rounded up with the introduction of the *attention mechanism* [Bahdanau et al., 2014]. Later on, inspired by the Hidden-Markov-Model (HMM) word-alignment model [Vogel et al., 1996] used in statistical machine translation , Yu et al. [2016] proposed a segment-to-segment architecture that learns to generate and align simultaneously. The alignment module extends the work of Graves [2012] and is capable of modeling local non-monotone mappings by allowing recurrent dependencies between monotone mappings. The idea was tested in mappings at the word level for the task of abstractive summarization, and in mappings at the character level for the task of morphological inflection.

More recent efforts have proposed combined strategies to bring together finite states machines and neural architectures in a more explicit way. One line of research replaces

hand-engineered features in the scoring function of a WFST with path scores obtained with a recurrent neural network (RNN, Rastogi et al. [2016], Lin et al. [2019]). In contrast, Schwartz et al. [2018] proposed SoPa, an end-to-end neural transducer with the same theoretical expressive power of linear-chain weighted finite state automata (WFSA). SoPa, for *Soft Patterns*, draws principles from one-layer convolutional neural networks (CNNs) in order to support flexible lexical matching [Davidov et al., 2010]. The architecture implements the state-transition function as a transition matrix that processes input one step at a time, like an RNN. The model is tested in text classification tasks including sentiment analysis, showing impressive robustness in low resource scenarios.

This connection between RNNs and CNNs with WFSA is later formalized by Peng et al. [2018]. They lay out theoretical proof that the recurrent hidden state update of a restricted set of RNNs is equivalent to the forward calculation of a weighted finite state automaton. Peng et al. [2018] defined such recurrence updates as *rational recurrences*.

2.2 Morphological String Transduction under Low-Resource Scenarios

In this section, we survey lines of research related to morphological string transduction tasks, namely inflection generation, paradigm completion, and lemmatization. We start by reviewing past editions of the SIGMORPHON Shared Tasks [Cotterell et al., 2016, 2017, 2018, McCarthy et al., 2019] and follow up with independent efforts in the literature.

The number of featured languages in the SIGMORPHON Shared Tasks has significantly increased from 10 languages (with one dataset per language) in its first edition [Cotterell et al., 2016] to 66 languages (with more than 100 datasets in total) in its last edition [McCarthy et al., 2019]. The editions of 2017 and 2018 [Cotterell et al., 2017, 2018] featured experimental scenarios with increasingly limited resources (high, medium, low), a.k.a. data regimes, for the task of type-level (i.e. context agnostic) inflection in order to investigate the generalization capability of the submitted systems under low-resource scenarios. The 2019 edition [McCarthy et al., 2019] introduced a slightly different setup to type-level inflection, this time with only a low-regime dataset for a target language but accompanied by a high regime dataset of a support language (not necessarily related

but highly resourced). The 2019 edition also featured the task of lemmatization in context, i.e. with access to sentential information. Although a low regime was not explicitly stated in the task setup, several datasets indeed fall into the low-regime categorization, e.g. English PUD has only 800 and 100 sentences for training and testing, respectively.

Despite the impressive efforts laid out to tackle morphological string transduction in these shared tasks, doing so under low-resource settings remains a challenge. Among the lines of research focused on tackling the data sparcity problem, three main strategies can be identified.

The first strategy consists in learning to transduce input characters into a sequence of edit operations instead of a sequence of characters [Makarov and Clematide, 2018a, Schröder et al., 2018, Dumitrescu and Boros, 2018, Hauer et al., 2019]. The defined edit actions operate at the character level and are obtained from the output of the Levenshtein algorithm, an extended version of the edit-distance algorithm. However, these systems rely on pre-aligned ⟨lemma,inflection⟩ pairs.

The second proposed strategy was to deliberately bias the network into copying word forms. On the one hand, Zhou and Neubig [2017] proposed to augment the training data with synthetic data, namely *hallucinated* data, for the task of context-agnostic inflection generation. This augmentation method extends the original set of lemma–word form pairs with pairs of forms with the same lemma, i.e. pairs of forms in the same paradigm. On the other hand, Madsack and Weißgraeber [2019] tackled the problem as a domain adaptation approach. The model is first trained to copy word forms for several epochs and then 'fine-tuned' over actual inflection pairs during the last epochs.

The last identified strategy is related to the previous one, and consists of taking on a multi-lingual training strategy. Madsack and Weißgraeber [2019] combined data from low-resourced languages with data from related, highly resourced languages. Kondratyuk [2019], on the other hand, combined the data of all available languages and trained the model over a shared vocabulary. Even though both of them report impressive boosts in performance, it is still not clear whether any transfer learning is happening between languages or whether having more data further biases the model to copy token strings.

In parallel with the efforts on SIGMORPHON shared tasks, one line of research ex-

plored more restricted ways to align the input and output characters. In the context of morphological inflection, Aharoni and Goldberg [2016] proposed to use monotonic alignments (i.e. characters can not be aligned to previously seen characters as we go from left to right) as a proxy for hard attention. The architecture is modeled as a read-only Turing machine in which the reader's pointer is represented by an attention module that points to a single input at each time step and moves from left to right. Following the setup proposed by Yu et al. [2016], the transducer leverages the enriched representation of the input string to condition decoding one character at a time. However, the transducer relies on externally calculated character-level alignments using the method proposed by Sudoh et al. [2013]. Building upon this line of work, Makarov and Clematide [2018c] introduced the exploration of valid action sequences during training in order to mitigate the dependence on an external aligner. Performance is reported to be comparable to the state-of-the-art, if not superior, in several benchmarks for the tasks of inflection generation and lemmatization. This transducer is first warm-started following the training procedure proposed by Aharoni and Goldberg [2016]. Then, the model is optimized by minimizing the expected risk (Minimum Risk Training, MRT). This training approach, as mentioned in section 1.5.4 directly optimizes sequence-level performance metrics, e.g. the Levenshtein distance between the gold lemma and the final transformed form. In this scenario, Makarov and Clematide [2018b] followed an imitation learning approach and proposed an expert policy to obtain a completely end-to-end training procedure, and eliminating the need for an external aligner or pre-training. This model further outperforms its counterpart trained with MRT.

Our approach follows the core idea behind the work of Makarov and Clematide [2018c] with the crucial difference that the derived edit actions operate at the word level instead of the character level. In addition, we leverage a multi-lingual representation space for actions that allows the models to share inductive bias in high-resourced related languages, dramatically improving performance for the task of morphological tagging.

2.3 Morphological Tagging under Low Resource Scenarios

The 2019 edition of the SIGMORPHON shared task [McCarthy et al., 2019] featured the task of lemmatization and morphological tagging in context, i.e. given a sequence of word forms the goal is to tag each token with its lemma and corresponding morpho-syntactic description (MSD) label, also known as *feature bundle*.

The main approaches to tagging identified in the submissions differ in whether they predict the complete bundle, e.g. {`'N;NOM;Pl'`} [Kondratyuk, 2019, Üstün et al., 2019, Shadikhodjaev and Lee, 2019], or predict each atomic feature separately, e.g. {`'N'`, `'NOM'`, `'Pl'`} [Bhat et al., 2019, Straka et al., 2019]. As reported by the organizers, systems that predicted complete feature bundles suffered from data sparsity problems under low-resource scenarios, the issue being more acute for morphologically rich languages. In order to remedy this issue, Bhat et al. [2019] proposed a neural conditional random field model that predicted each morphological category (the 'dimensions' in UniMorph) in a hierarchical manner, starting with POS. Similarly, Straka et al. [2019] proposed to predict the label of each morphological category independently for each token, i.e. as many softmax layers as categories, in addition to predict the complete feature bundle.

Other lines of research have explored the impact of neural architectural choices not only across tagging strategies bu also across languages. For example, Heigold et al. [2017], proposed an LSTM-based tagger over character-based word representations and tested it on 14 languages of varying morphological richness. They compare RNN-based and CNN-based token representations and report that RNN representations are more robust than CNN in most cases. The best results, however, are achieved by implementing a voting system over several instances of both architectures, a strategy called *ensembling*.

In regards to the usage of word embeddings, Üstün et al. [2019] reports that pre-trained embeddings are better suited for morphological tagging, whereas end-to-end embeddings (i.e. trained from scratch) are better suited for lemmatization. More recent work has explored the benefits of contextualized word representations, such as ELMo [Peters et al., 2018] and BERT [Devlin et al., 2019], in the input representation [Kondratyuk, 2019, Üstün et al., 2019, Straka et al., 2019].

3 Transducing Pseudo Morphological Processes for Lemmatization and Morphological Analysis in Context

In this chapter we define our proposed edit-action set and elaborate on how they resemble morphological processes. Then, we investigate how this action set can be used to tackle the tasks of context-aware morphological tagging and lemmatization for a variety of languages that resort to different combinations of word formation processes during inflection.

3.1 Problem Formulation

Let $w \in V$ and $z \in V^L$ be a word type and its corresponding lemma; and let $\mathcal{A}$ be a set of transformation actions over strings. We define the function $T : V \times \mathcal{A}^m \mapsto V^L$ that receives as input a word form w and a sequence of string transformations $a = \langle a_0, ., a_i, .., a_m \rangle$. T iteratively applies the transformations one at a time and returns the resulting string. The objective is to obtain a sequence of actions a such that a form w gets transformed into its lemma z, i.e. $T(w, a) = z$.

3.1.1 String transformations at the word level

We encode every string transformation –henceforth, action– $a_i \in \mathcal{A}$ as follows:

⟨`operation-position-segment`⟩

where `operation` denotes the kind of transformation we want to perform, e.g. a deletion or insertion. The `position` component indicates where the operation should be performed (zero-indexed, measured from the left border of the token). Finally, the `segment` component provides the sequence of contiguous characters involved in the operation, e.g. which characters to be inserted. Table 3 presents a description of the licensed values of each component, including the operation set considered. Note that operation 'transposition' denotes the swap in positions of two segments, i.e. `AB` → `BA`. Note that, by definition, the segments involved can be longer than one character.

Consider the example presented in Table 4, a sequence of suffix actions. The form *visto* (Spanish for 'seen', past participle) is transformed into the lemma *ver* ('to see'), with all actions operating at the right border of the current token. Each action modifies

Component	Label	Description
`operation`	INS	insert
	DEL	delete
	SUBS	substitute
	TRSP	transpose
	STOP	stop
`position`	_A	at the beginning (prefix)
	A_	at the end (suffix)
	_i_	at position i
`segment`	c	$c \in \Sigma^* \backslash \{\emptyset\}$

Table 3: Description of components encoded in action labels. Σ: alphabet of set of characters observed in the training data.

Token	Action
visto	DEL-A_-o
vist	DEL-A_-t
vis	SUBS-A_-er
ver	STOP
visto	`DEL-A_-o DEL-A_-t SUBS-A_-er STOP`

Table 4: Example of step-by-step transformation from form *visto* (Spanish for 'seen', past participle) to lemma *ver* ('to see'). Bottom row presents the final token representation as the initial form followed by the action sequence.

the current form of the word, the next action operates over this modified form, and so on and so forth.

3.1.2 Obtaining gold action sequences

We discuss now how to deterministically populate $\mathcal{A}$. We start off with operations that act upon one character at a time. We obtain these operations with the Damerou-Levenshtein (DL) distance algorithm which adds the 'transposition' operation in addition to the traditional set of the edit-distance algorithm. Using the word-level format introduced in the previous section, the DL algorithm gives us actions of the form ⟨`operation-_i_-character`⟩, i.e. actions with numeric positions and single-character segments. A transducer model learning to perform actions like this would face serious sparsity issues. Hence, in order to tackle this sparsity, we simplify the set of values encoded as position in a two-steps process.

First, we merge the k most frequent operations performed at adjacent positions by

applying the byte-pair-encoding (BPE,Gage [1994]) algorithm over the initial sequences of sparse actions. The idea is to discover actions performed over contiguous characters instead of discovering subword units. For example, actions `DEL-4-e` and `DEL-5-d` would be merged into `DEL-4-ed`. Note that the position of the left-most action (position-wise) is preserved in the merged action.

Second, we replace the `position` component of actions performed at the beginning of a token with the label `_A`, indicating that it is a prefixing action. Analogously, we use the label `A_` to indicate it is a suffixing action. For example, the suffixing action `DEL-4-ed` would be transformed into `DEL-A_-ed`.

We also take into consideration the order in which these word-level actions should take place during transduction. Action sequences are sorted so that prefix actions (with position component `_A`) are performed first, followed by inner-word actions (positions `_`i`_`), and lastly, suffix actions (with position component `A_`). In addition, prefix and suffix actions are sorted so that T would process the word form from the outside in. Consider the example in Table 4, suffix action `DEL-A_-o` comes before action `DEL-A_-t` as it modifies the right-most segment in the word. This way of processing ensures that continuous strings, i.e. without gaps, are obtained as intermediate word forms at every step.

3.2 Lemmatization using action sequences

We posit the task of lemmatization as a language modeling problem over action sequences. Let $w = \langle w^0, \ldots, w^i, \ldots, w^n \rangle$ be a sequence of word tokens, $z = \langle z^0, \ldots, z^i, \ldots, z^n \rangle$ the lemma sequence associated with w, and $a^i = \langle a^i_0, \ldots, a^i_j, \ldots, a^i_m \rangle$ the action sequence such that $T(w^i, a^i) = z^i$. We encode a^i using an RNN with an LSTM cell [Hochreiter and Schmidhuber, 1997], as follows $\quad h^i_j = LSTM(e^i_j, h^i_{j-1})$
where e^i_j is the embedding of action a^i_j. Then, the probability of action a^i_j is defined as

$$P(a^i_j | a^i_{<j}; \Theta) = softmax(g(W * h_j + b)) \tag{8}$$

where $g(x)$ is the ReLU activation function, and W and b are network parameters. As a way to introduce the original word form into the encoded sequence, we insert w^i at the beginning of sequence a^i. Hence, the probability of the first action is determined by

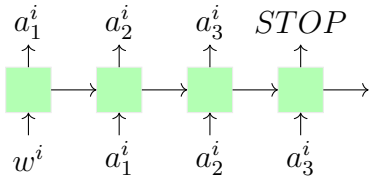

Figure 1: Architecture of LEM, our proposed lemmatization model posited as a language model over action sequences.

$h_0 = LSTM(w^i, h_m^{i-1})$ where h_m^{i-1} is the last state of the encoded action sequence of the previous word w^{i-1}.

The network is then optimized by minimizing the negative log-likelihood of the action sequences, as follows,

$$\mathcal{L}(W, \theta) = -\sum_{\langle w,z \rangle \in \mathcal{T}} \sum_{i=0}^{n} P(w^i|\theta) \cdot \tag{9}$$

$$\sum_{j=1}^{m} P(a_j^i | a_{<j}^i, \theta) \tag{10}$$

where $\mathcal{T}$ is the set of all token–lemma sentence pairs in the training set and θ represents the parameters of the network. Equation 10 is equivalent to obtaining the maximum likelihood estimate (MLE) over the training data. For this reason, we call this model Lem_{MLE}. Figure 1 presents an overview of the architecture. Note that a_m^i is the special action label $STOP$. During decoding, Lem_{MLE} receives as input sentence w and predicts an action sequence $\hat{a}^i$ for each token, from which the predicted lemma $\hat{z}^i$ is reconstructed by running T over $\hat{a}^i$.

In addition, we define the action space over which P in Equation 8 operates as the union of the action set $\mathcal{A}$ and the types vocabulary $\mathcal{V}$, i.e. $a_j^i \in \mathcal{A} \cup \mathcal{V}$. This way, the model has the chance to choose another word form as next action instead of replacing the string character by character.

3.3 Minimum Risk Training for Lemmatization

We formalize now the idea of introducing metric-based error optimization for lemmatization. Let $\Delta(\hat{z}^i, z^i)$ be a risk function that quantifies the discrepancy between the predicted lemma $T(w^i, \hat{a}^i) = \hat{z}^i$ and gold lemma z^i. Then, the model is trained by minimizing the expected risk, defined as

$$\mathcal{R}(\mathcal{T}, \Theta) = \sum_{\langle w,z \rangle \in \mathcal{T}} \sum_{i=0}^{n} \mathbb{E}_{a|w^i;\Theta}\big[\Delta(\hat{z}, z^i)\big] \tag{11}$$

where $\mathcal{T}$ is the training set and Θ represents parameters of the network. We use the risk function proposed by Makarov and Clematide [2018c], defined in terms of normalized Levenshtein distance (NLD) and accuracy, as follows

$$\Delta(\hat{z}, z^i) = NLD(\hat{z}, z^i) - \mathbb{1}\{\hat{z} = z^i\} \tag{12}$$

where $NLD(\hat{z}, z^i)$ is defined as the number of Levenshtein distance operations required to transform $\hat{z}$ into z^i, divided by $|\hat{z}|$. Accuracy $\mathbb{1}\{\hat{z} = z^i\}$ takes the value of 1 only when $\hat{z} = z^i$, and 0 otherwise.

As discussed in section 1.5.4, loss function $\mathcal{R}$ is intractable and has to be approximated by sampling action sequences from search space $\mathcal{A}^m$, as proposed by Shen et al. [2015]. Hence, the expectation of the risk under the posterior distribution $P(a|w^i;\theta)$ in Equation 11 is approximated by

$$\mathbb{E}_{a|w^i;\Theta} \approx \sum_{a \in S(w^i)} Q(a|w^i;\Theta,\alpha)\Delta(\hat{z}, z^i) \tag{13}$$

where $S(w^i) \subset \mathcal{A}^m$ is a sampled subset of the search space of possible action sequences for w^i. The distribution $Q(a|w^i;\Theta,\alpha)$ is defined on the subspace $S(w^i)$ and has the form

$$Q(a|w^i;\Theta,\alpha) = \frac{P(a|w^i;\Theta)^\alpha}{\sum_{a' \in S(w^i)} P(a'|w^i;\Theta)^\alpha} \tag{14}$$

where $\alpha \in \mathbb{R}$ is a hyper-parameter that controls the sharpness of the distribution. We name a model *Lem* trained to minimize risk $\mathcal{R}(\mathcal{T}, \Theta)$ as Lem_{MRT}.

3.4 Morphological Tagging

Given the sequence of word tokens $w = \langle w^0, \ldots, w^i, \ldots, w^n \rangle$, the task consists on tagging each token with a morpho-syntactic description (MSD) label $F^i = \{f_0^i, \ldots, f_k^i, f_K^i\}$, where F^i is the concatenation of all individual features f_k such as N or Pl.

Our tagging framework consists of two main components, a hierarchical encoder that encodes action sequences into word-level representations, and a word-level MSD tagger.

We first elaborate on the architecture of the hierarchical encoder and then propose two tagger variants that operate on top of it, namely a tagger that predicts the MSD bundles F^i and a decoder tagger that predicts each f_k^i in sequence.

3.4.1 Hierarchical Action Encoder

The first component of our model is the hierarchical encoder which encodes action sequences into word representations. Formally, given the action sequence $a^i = \langle a_0^i, \ldots, a_j^i, \ldots, a_m^i \rangle$ associated with token w^i, we start by encoding a^i using a bidirectional LSTM Graves et al. [2013] as follows,

$$f_j = LSTM_{fwd}(a_j^i, f_{j-1})$$
$$b_j = LSTM_{bwd}(a_j^i, b_{j+1})$$

where $LSTM_{fwd}$ and $LSTM_{bwd}$ are the forward and backward cells, respectively. Then, token w^i is represented by $x^i = [f_m; b_0]$, where f_m is the the last forward state and b_0 is the first backward state. Afterwards, word level representations $x^0, .., x^n$ are further encoded using another bidirectional LSTM layer in order to enrich each token representation with context from both sides of the sentence. This way, we obtain $u^i = biLSTM(x^i, u^{i-1})$ (forward and backward output concatenated) as word level representations that are passed down to the next component of the model. Figure 2 presents the architecture of the hierarchical encoder. Note that the action encoder is initialized with the last hidden state of the previous encoded action sequence, c^{i-1}. This way, the action encoder is aware of actions predicted for previous word tokens.

3.4.2 MSD Bundle Tagger

The first sequence tagger proposed is named MBundle and it predicts complete MSD label bundles instead of fine-grained feature labels. Formally, given word-level representation u^i, the probability of feature label F^i is given by

$$p(F^i | x^{1:i-1}, \theta) = softmax(g(W * u^i + b)) \tag{15}$$

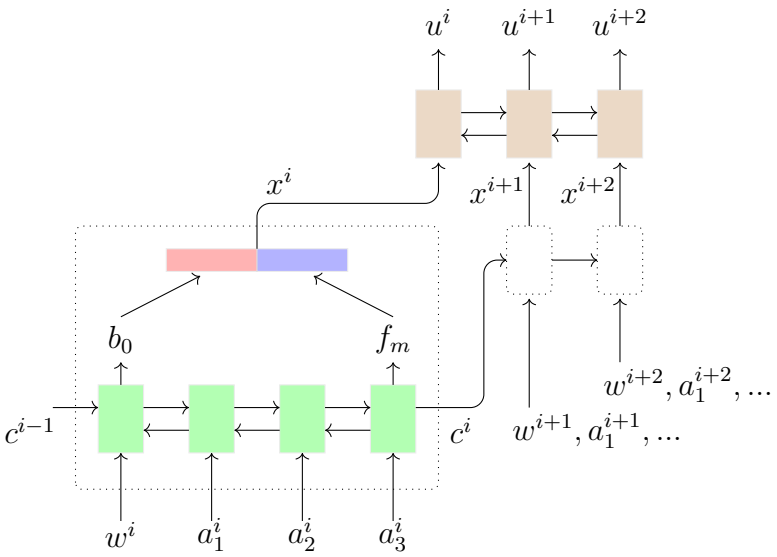

Figure 2: Architecture of the hierarchical action encoder component of our morphological tagger models.

where $g(x)$ is a ReLU activation function, and W and b are network parameters. The network is optimized using cross-entropy loss. Figure 3 presents an overview of the architecture of this model.

3.4.3 Fine-grained MSD Tagger

Our second proposed tagger, named MSeq, relies on an encoder-decoder architecture to predict fine-grained MSD labels in sequence, one at a time. The decoder is a unidirectional LSTM extended with a global attention mechanism with general score function [Luong et al., 2015]. Formally, given the decoder side hidden state h_k^i and a encoder side context vector d_k^i, the attention-enriched decoder hidden state is defined as $\hat{h_k^i} = W_d[d_k^i; h_k^i]$ [5].

Then, the probability of fine-grained MSD label f_k^i is defined by

$$p(f_k^i | f_{<k}^i, u^i) = softmax(W_s \hat{h_k^i} + b_s) \tag{16}$$

where u^i is the token representation provided by the hierarchical action encoder, and W_s and b_s are network parameters. Figure 4 presents an overview of the architecture of this model.

[5] We employ plain linear combination instead of a *tanh* activation (used by Luong et al. [2015]) since it produced better results in preliminary experiments.

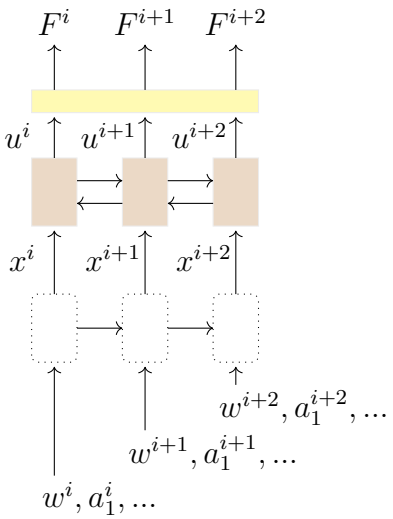

Figure 3: Architecture of the MBundle morphological tagger.

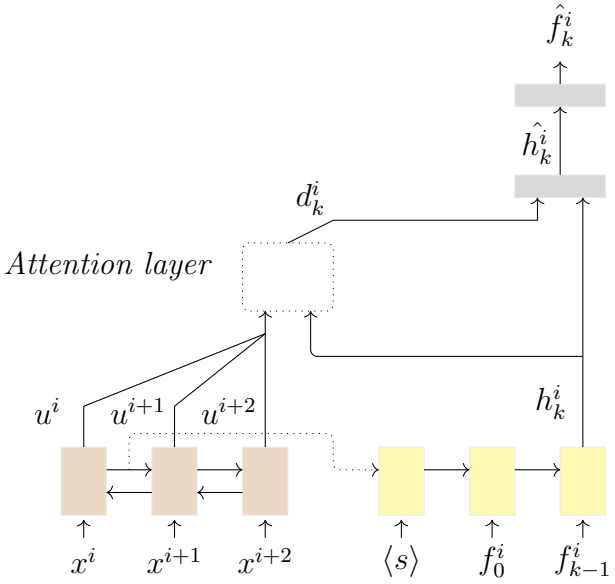

Figure 4: Architecture of the MSeq morphological tagger. Encoding of actions into x^i are omitted for simplification.

3.4.4 Tagging over multilingual actions

The action sequences obtained with the method described in section 3.1.2 are language-dependent. Hence, the variety of actions learned is limited to the word formation preferences attested for a specific language and how well represented the inflection paradigms are in the training data. We tackle this limitation by taking advantage of the arguably universal and language-agnostic notion of word formation processes and how they can signal

morpho-syntactic phenomena. However, one must remain wary that a specific morpho-syntactic phenomenon might be signaled by different types of word formation processes across languages. Consider the verb 'to like' and the morpheme for verb negation (in *italics*) in the following example:

(1) a. English: *dis-* like

b. Spanish: *dis-* gustar

c. Turkish: been *-me* -mek

Even though this morpho-syntactic phenomenon is signaled by prefixation in English and Spanish, it is signaled by suffixation in Turkish.

We experiment with projection of action embeddings from a variety of languages into a common embedding space. Thus, a morphological tagger can take advantage of the common word formation patterns encoded in a language-agnostic space and how they signal morpho-syntactic phenomena. Since it is unrealistic to collect annotated data featuring how a morphological phenomenon is realized in several languages, we resort to the unsupervised projection method proposed by Lample et al. [2018b].

We name actions embeddings derived this way MULTI-ACTION. Bear in mind, however, that each lemmatizer is language-specific. Hence, during decoding a tagger will query the language-specific lemmatizer, obtain a sequence of actions and then use the multilingual embeddings of these actions as input.

4 Experimental Setup

In this chapter we investigate the effectiveness of our proposed models for the tasks of lemmatization and morphological analysis in context. All models were implemented and trained using PyTorch v1.0.0. [6]

Our experiments follow the setting of the SIGMORPHON 2019 Shared Task on 'Cross-linguality and Context in Morphology' [McCarthy et al., 2019] at which early experiments were submitted [Cardenas et al., 2019]. We release the code of our proposed lemmatization and tagging models.[7]

4.1 Datasets

We experiment with the official treebank splits (train, dev, and test) for Shared Task II [McCarthy et al., 2019].[8] These treebanks are re-splitted versions of the UD treebanks v.2.3 [Nivre et al., 2018] with feature bundles translated from UD's UFEAT tagset into the UniMorph tagset [Kirov et al., 2018] using the mapping strategy proposed by McCarthy et al. [2018]. We consider the following languages and treebanks: English (en_ewt), Spanish (es_ancora), Turkish (tr_imst), Czech (cs_pdt), German (de_gsd), and Arabic (ar_padt). Table 5 presents the statistics of the training sets for all languages.

Language	Num. sents.	Num. tokens	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$
en	13,297	204,857	17,342	282
es	14,144	439,925	34,912	479
cs	70,330	1,207,922	113,932	872
tr	4,508	46,417	14,645	675
ar	6,131	225,494	22,478	617
de	27,628	536,828	43,188	720

Table 5: Corpus statistics of training splits for all languages considered. Num. sents: number of sentences; $\lvert\mathcal{V}\rvert$: size of types vocabulary; $\lvert\mathcal{A}\rvert$: size of the action set.

[6] `https://pytorch.org/`
[7] `https://github.com/ronaldahmed/morph-bandit`
[8] `https://github.com/sigmorphon/2019/tree/master/task2`

4.2 Action sequence preprocessing

We lowercase forms and lemmas before running the DL-distance algorithm. Following the BPE training procedure described by Sennrich et al. [2016], we obtain the list of merged operations from the action sequences derived from the training data. We limit the number of merges to 50. Then, these merges are applied to action sequences on the development and test data. Table 5 presents the size of the derived action set per language.

4.3 Baseline models

We consider the baseline neural model provided by the organizers of the SIGMORPHON Shared Task. The architecture, proposed by Malaviya et al. [2019], performs lemmatization and morphological tagging jointly. The morphological tagging module of the model employs an LSTM-based tagger [Heigold et al., 2017], whilst the lemmatizer module employs a sequence-to-sequence architecture with hard attention mechanism [Xu et al., 2015]. We refer to this model as Base.

4.4 Evaluation Metrics

We consider the following evaluation metrics, regarded as standard in the literature.

- Lemmata accuracy: 0|1 accuracy of lemmata, i.e. whether the predicted string is exactly the same as the gold string.
- Average Levenstein distance of lemmata: Levenstein distance between predicted and gold lemmata, not normalized by length, averaged over all lemmas and sentences in the test set.
- MSD Accuracy: 0|1 accuracy of morpho-syntactic description bundle labels.
- F1 score for MSDs: Micro-averaged over individual, fine-grained feature labels.

4.5 Lemmatization with MLE objective

The Lem_{MLE} model is optimized using Adam [Kingma and Ba, 2017] and regularized using dropout [Srivastava et al., 2014] over 20 epochs. Training is halted if the loss

Hyper-parameter	Lem_{MLE}	MBUNDLE
Batch size	128	24
Learning rate	6.90E-05	1.00E-04
Dropout	0.19	0.05
Epochs / patience	20 / 5	100 / 30
Action embedding	140	140
Action-LSTM cell	100	100
Word-LSTM cell	-	100
FF layer size	100	100

Table 6: Hyper-parameters of lemmatization model Lem_{MLE} and tagging model MBUNDLE.

over the validation set does not decrease after 5 epochs, i.e. following an early stopping strategy. We tune the hyper-parameters of both models over the development set of Spanish (es_ancora)[9] and then we use the optimal configuration to train on all languages. The hyper-parameters were optimized over 30 iterations of random search guided by a Tree-structured Parzen Estimator (TPE).[10] Table 6 presents a summary of the optimal hyper-parameters found.

During decoding, we use temperature to smooth the probability distribution of the next action $P(a_j|a_{<j};\theta)$. Formally, given a temperature τ, the distribution in Equation 8 on page 22 becomes

$$P(a_j|a_{<j};\theta) = softmax(g(W * h_j + b)/\tau) \tag{17}$$

In this setup, we perform decoding using a greedy decoder with temperature of 1. We also experimented with beam search decoding but the improvements were not significant. Furthermore, we implement heuristics to prune a predicted sequence of actions. In addition to the heuristic of halting decoding if a PAD or STOP action is found, we halt if the action is not valid given the current string. For example, the action `DEL-_5_-o` cannot be applied to string `who` for the simple reason that the string is not long enough and, hence, the action is not valid.

[9]We wanted to use a language that is morphologically more complex than English as our reference.
[10]We use HyperOpt library (`http://hyperopt.github.io/hyperopt/`)

Hyper-parameter	Optimal value
Batch size	5
Learning rate	1.00E-4
Q sharpness smoothing (α)	1.00E-4
Sample size ($\|S(w^i)\|$)	20
Temperature	1

Table 7: Hyper-parameters of lemmatization model Lem_{MRT}. Architectural hyper-parameters are the same as for Lem_{MLE}.

4.6 Lemmatization with MRT

The Lem_{MRT} model is optimized using Adadelta [Zeiler, 2012]. Preliminary experiments showed that training converged slower and in some cases diverged when optimizing with Adam. Training is set up with a warm start by initializing the model with the corresponding Lem_{MLE} model, using a batch size of 5 and learning rate of $1e^{-4}$. All other architecture hyper-parameters are set to the same value as for Lem_{MLE} models. Following the procedure described by Shen et al. [2015], we sample a fixed number of actions sequences and discard the repeated ones. Also, we include the gold action sequence in the final sampled set.

In addition, we analyze the effect of hyper-parameters exclusive to the MRT setup such as the sharpness smoothing parameter α, subsampled subset size, and temperature during decoding. The fine-tuning of hyper-parameters in this section were performed over the Spanish (es_ancora) validation set and measured in terms of lemmata accuracy and Levenshtein distance. The optimal values for these hyper-parameters are presented in Table 7 for convenience.

4.6.1 Effect of Q sharpness smoothing (α)

The parameter α controls the sharpness of distribution Q (see Equation 14). Figure 5 presents the effect of α when using a sample size of 20 and temperature of 1 during decoding. We observe that higher values of α tend to destabilize training and cause metrics values to worsen at later epochs. A value of $alpha = 1e^{-4}$ is observed to lead to consistently more stable training and better performance. We also tested $alpha = 1e^{-5}$ but training time increased notably and performance did not improve significantly. Hence, we set $alpha = 1e^{-4}$ for all following experiments.

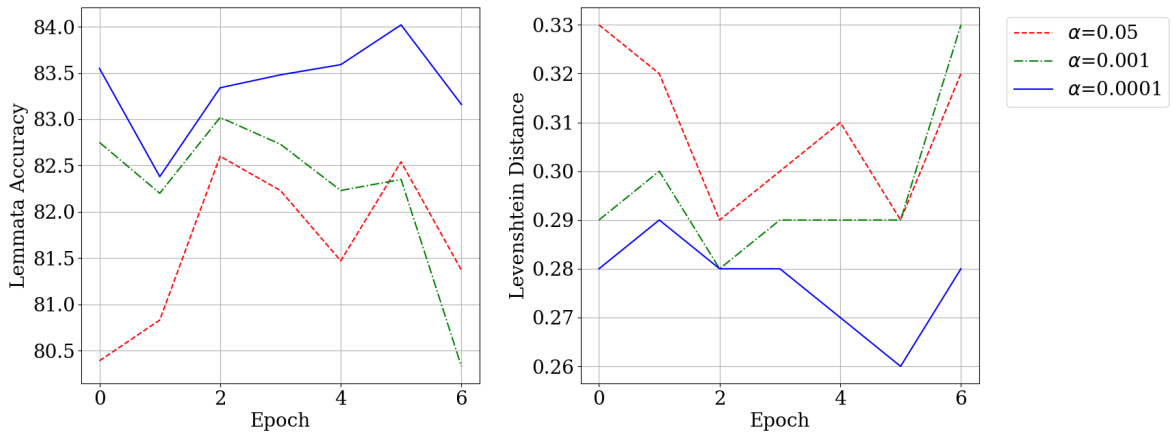

Figure 5: Effect of sharpness smoothing (α) on Lem_{MRT} as measured by lemmata accuracy (left) and Levenshtein distance (left) for the Spanish (es_ancora) validation set.

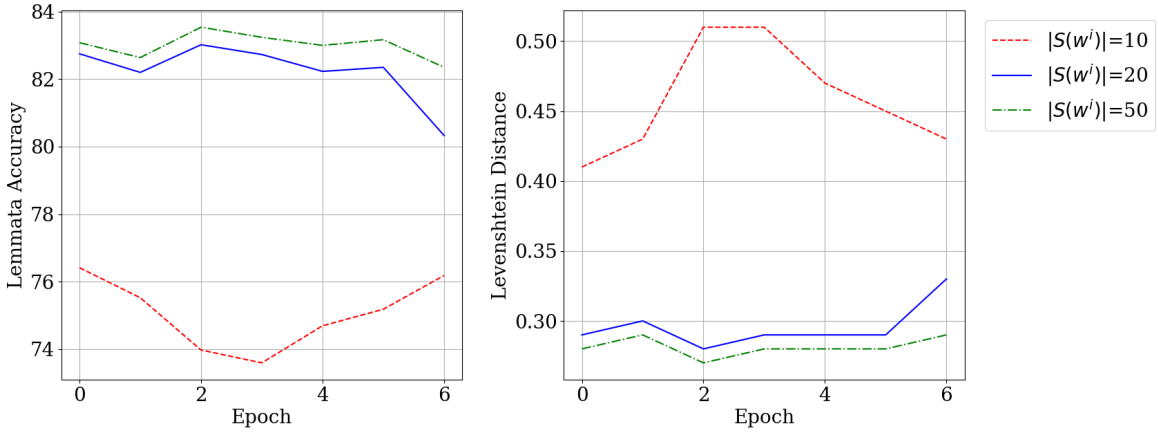

Figure 6: Effect of sample size ($|S(w^i)|$) on Lem_{MRT} as measured by lemmata accuracy (left) and Levenshtein distance (left) for the Spanish (es_ancora) validation set.

4.6.2 Effect of sample size

As presented in Equation 13, the quality of approximation of posterior distribution $P(a|w^i;\theta)$ by Q depends on the size of the subsampled space $S(w^i)$. As shown in Figure 6, performance consistently improves as the sample size increases. This expected behavior comes with a training time trade-off. A sample size of 50 makes training three times slower w.r.t. size 20, and a sample size of 100 makes it six times slower. Moreover, we observed no significant improvement for sample sizes greater than 20. Hence, we use a sample size of 20 for following experiments for efficiency.

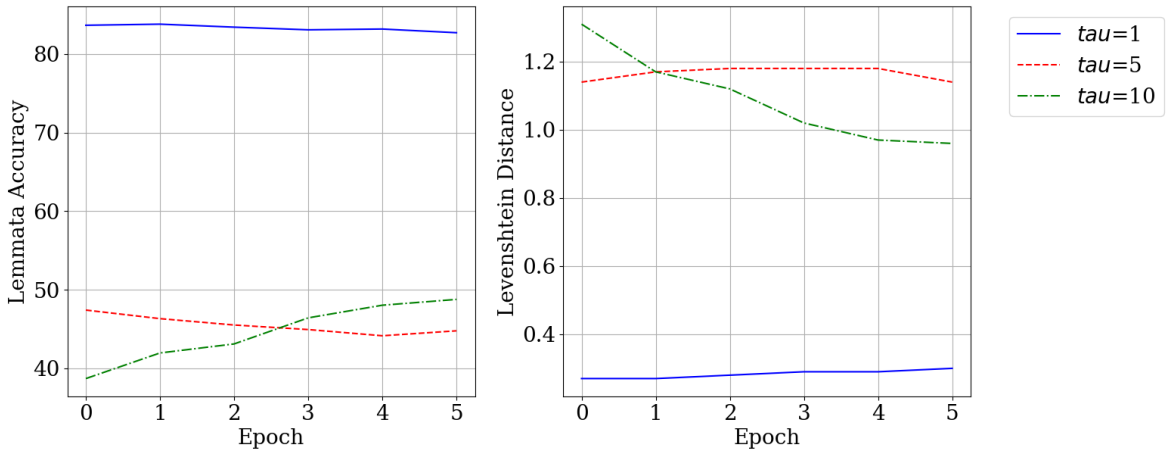

Figure 7: Effect of decoding temperature (τ) on Lem_{MRT} as measured by lemmata accuracy (left) and Levenshtein distance (left) for the Spanish (es_ancora) validation set.

4.6.3 Effect of temperature during decoding

We also investigate how temperature influences diversity during decoding and how it impacts performance. We observe that probability distribution $P(a_j^i|a_{<j}^i;\theta)$ in Equation 8 is heavily biased towards producing short sequences. This is highly desirable for highly fusional or inflected languages since they usually present one-slot morphology, e.g. Spanish. In Figure 7, we observe that increasing the temperature hurts performance. This is to be expected as a higher temperature smooths the spikiness of P and forces the model to pick otherwise less probable actions, which in turn leads to longer sequences. Hence, we use a temperature of 1 for following experiments.

4.7 Morphological Tagging models

We initialize action embeddings of the hierarchical action encoder with embeddings learned with *Lem* models and let the tagger fine-tune them during training. Both taggers, MBUNDLE and MSEQ, are optimized using Adam. For MSEQ, we use an LSTM decoder cell of size 100 and a maximum length of decoded feature sequence of 25. The following embedding-tagger combinations were investigated.

- Lem_{MLE}–{MBUNDLE,MSEQ}. Taggers are initialized with monolingual Lem_{MLE} embeddings.

- MULTI-MBUNDLE. Tagger is initialized with multilingual action embeddings MULTI-ACTION. We project MLE-trained action embeddings with 5 iterations of Procustes refinement. All projections were made into the Spanish embedding space.[11]

4.8 Co-occurrence of actions and morphological features

We investigate the co-occurrence of action labels with individual morphological features. Given the word form w^i and its associated morphological tag $F^i = \{f_0^i, ..., f_k^i, f_K^i\}$ and action sequence $a^i = \langle a_0, ..., a_j, ..., a_m \rangle$, let us define the joint probability distribution between individual features and action labels, as

$$p(f_k^i, a_j^i) = P(f_k^i | x_{1:i}) \cdot P(a_j^i | a_{1:j-1}^i) \tag{18}$$

We consider $P(F^i|x_{1:i}) = P(f_k^i|x_{1:i}), \forall f_k^i \in F^i$. Note that
$P(F^i|x_{1:i})$ and $P(a_j^i|a_{1:j-1}^i)$ are the probabilities obtained by the lemmatizer and tagger in equations 8 and 15, respectively.

4.9 The SIGMORPHON Shared Task II

Past editions featured tasks like type-level inflection and context-aware re-inflection [Cotterell et al., 2016, 2017, 2018], most notably increasing the number of languages in the analysis from 40 in 2017 to 66 in this last edition.[12]

We focus on Task II 'Morphological Analysis and Lemmatization in Context' 2019, where early results where submitted. Given a tokenized sentence, we must predict the lemmas and MSD labels for each word. We participated under the name `CHARLES-MALTA-01`. The system submitted corresponds to the lemmatizer-tagger combination Lem_{MLE}-MBUNDLE. All treebanks were trained using the optimal hyper-parameters listed in Table 6 except for Komi Zyrian (kpv_ikdp, kpv_lattice) and Sanskrit (sa_ufal), for which we observed instable behaviour during training. Hence, we trained the MBUNDLE tagger over treebanks kpv_ikdp, kpv_lattice, and sa_ufal with batch size of 40, learning rate of 0.01, dropout of 0.07, action encoder cell of size 10, word encoder cell of size 40, and a gradient clipping threshold of 0.38.

[11] Preliminary experiments showed that projected MLE-trained embeddings led to better tagging performances w.r.t. projected MRT-trained embeddings.

[12] At the time of writing, SIGMORPHON 2019.

5 Results and Discussion

Each lemmatization and morphological tagging model was trained 10 times, each time initialized with a different random seed. The evaluation results in this chapter are presented as the average over 10 runs.

In addition, we conduct statistical significance tests when comparing our models. We use the *bootstrap test* [Efron and Tibshirani, 1986], a non-parametric test, since our metrics are not normally distributed. We follow the implementation proposed by Berg-Kirkpatrick et al. [2012] with a p-value threshold of 0.05.

5.1 Lemmatization

Table 8 presents lemmatization performance of the training objectives tested on our architecture, as measured by lemmata accuracy (LAcc) and Levenshtein distance (Lev-Dist). We observe mixed results across languages when optimizing using MRT. Relative error increase in lemmata accuracy (LAcc) ranges from non-significant (0.11%) for *en* to 4.9% for *de* and 5.97% for *es*. In contrast, we observe a relative error decrease ranging from non-significant (0.53%) for *cs*, to 6.12% for *tr* and up to an encouraging 55.73% for *ar*.

We hypothesize that the relative poor performance stems from the input representation, i,e. the action sequences. Recall from Section 3.1.1 that an action label encodes the operation to perform (e.g. delete), where to perform this operation (e.g. at end of the word), and the character segment involved (e.g. -*s*). We limit ourselves to predict action labels attested in the training data, namely the action space $\mathcal{A}$, since the combination of all possible options to encode in an action label can grow exponentially. Nevertheless, we find that the encoded position (`_i_`) and the character segment induce an action space $\mathcal{A}$ that is too fine-grained and sparse, even after the BPE merging of adjacent actions. We now elaborate on how the size of the action space impacts lemmatization performance.

The results suggest that MRT harms performance when the complete search space, $\mathcal{A} \cup \mathcal{V}$, is so large that the subsampled space cannot appropriately represent the sparse, original search space. Consider the following two cases: (i) *cs*, with a search space size of 114804, and (ii) *tr*, with 47092 (see Table 5). In terms of Levenshtein distance, minimizing risk for *cs* induces an error increase of 7% w.r.t. maximization of likelihood. However,

MRT does improve over MLE training for *tr* with a 11.62% error reduction in terms of Levenshtein distance. We observe similar trends in other highly inflected languages like *es* and *de* for case (i), and in *ar* for case (ii).

Moreover, we find that the performance gap, as measured by accuracy score, can be lessened or even slightly reverted by using more training data. This is the case of *cs* for which the training set is the largest in our study (see Table 5). We also observe that MRT is most effective in terms of accuracy for *tr* and *ar*, despite having much less training data than the other languages. This could be due to their relatively small type vocabulary which makes sampling the complete search space much more effective.

Language	Lem_{MLE}		Lem_{MRT}	
	LAcc	Lev-Dist	LAcc	Lev-Dist
en	89.36	0.15	89.28	0.16
es	84.88	0.24	83.58	0.28
cs	86.13	0.26	86.59	0.28
tr	64.75	1.29	68.73	1.14
ar	44.12	1.49	68.71	1.02
de	68.35	0.45	65.00	0.70

Table 8: Lemmatization performance under MLE training (Lem_{MLE}) and MRT (Lem_{MRT}) over test sets. LAcc: lemmata accuracy; Lev-Dist: levenshtein distance.

We further assess the performance of our models in ambiguous cases, i.e. when a word form may be associated with more than one lemma but only one is correct given the context. We follow the experiment design proposed by McCarthy et al. [2019] and distinguish between the following word types categories: ambiguous (more than one lemma in the training set), unseen, seen unambiguous (only one lemma), and all. Figure 8 presents relative improvement scores of accuracy per category for all languages analyzed. In general, we observe that MRT heavily harms performance over unseen forms for all languages except *tr*, for which a slight improvement is observed. For *ar*, it is worth noting that even though MRT leads to a ∼50% error increase for unseen forms, it also leads to an error decrease of more than 30% in all other categories. Besides *ar*, *tr* is also benefited by MRT on ambiguous cases with an error decrease of ∼13%.

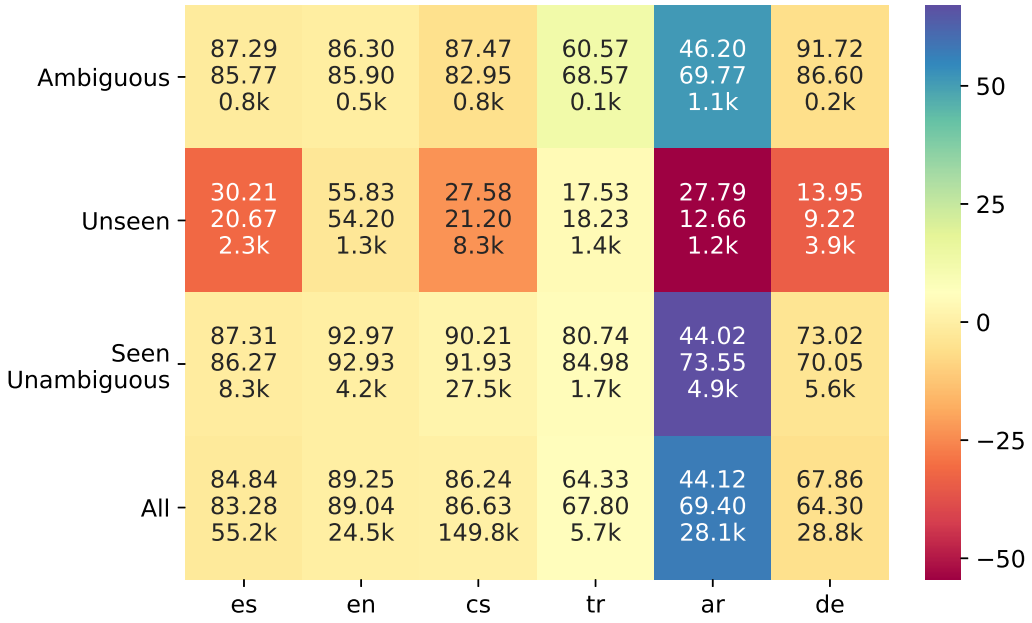

Figure 8: Performance by type of inflected form over the development set of all languages. In each cell, color indicates relative improvements of Lem_{MRT} (middle row score) over Lem_{MLE} (top row score), as well as the respective number of tokens (bottom row).

5.2 Morphological Tagging

Table 9 presents the results on morphological tagging for the lemmatizer-tagger model combinations investigated. First, we observe that the MSeq tagger underperforms MBundle in all languages except *en* and *tr*. Upon closer inspection, we noticed that the annotation of MSD labels was not consistent with UniMorph guidelines [Sylak-Glassman, 2016] regarding the order of dimensions, e.g. both labels `N;SG;MASC` and `MASC;N;SG` are present in the *en* training set. This scenario will definitely prevent a decoder-based tagger like MSeq from learning a meaningful order of labels effectively. The improvement we observe for *en* and *tr* might be due to the more careful annotation and consistency of MSD labels w.r.t. other languages.

Second, we observe substantial improvement by using multilingual action embeddings in all languages (Multi-MBundle), ranging from 6.87% (*de*) to 19.68% (*en*) in absolute F1-score. After *en*, *cs* is the most benefited language. This might be due to our decision

Language	Lem_{MLE}-MBundle		Lem_{MLE}-MSeq		Multi-MBundle	
	MAcc	M-F1	MAcc	M-F1	MAcc	M-F1
en	62.80	70.38	67.29	80.55	88.37	90.07
es	72.60	78.76	49.23	67.49	87.31	89.65
cs	63.13	76.45	34.10	64.25	83.93	89.14
tr	25.76	42.14	27.43	45.35	50.84	54.26
ar	51.77	62.52	28.82	56.11	61.28	70.46
de	58.10	72.91	37.56	53.94	68.49	79.78

Table 9: Results on morphological analysis of proposed models over the test set. MACC: MSD accuracy; M-F1: MSD micro-F1 score.

of having the *es* action space as target for embedding projection. A language that marks a great deal of morphological phenomena, such as *es*, will have a richer and more granular action space compared to a language that –almost– does not mark morphological phenomena such as *en*. Hence, a rich and granular target action space benefits projection from action spaces with similar level of granularity, since fewer information will be lost.

5.3 SIGMORPHON 2019 submission

Table 10 presents performance of our submission according to all metrics for the top 5 and bottom 5 scored treebanks according to the MSD-F1 scores on the official test evaluation. Please refer to Appendix A.1 for results on all languages. In general, our model underperforms the baseline for most treebanks. In lemmatization, we observe an error increase ranging from 0.27% to 35.14% in lemma accuracy. However, we improve over the baseline on the following languages: Tagalog (tl_trg), Chinese (zh_gsd, zh_cfl), Cantonese (yue_hk), and Amharic (am_att).

In morphological tagging, we observe an error increase ranging from 0.31% to 7.34% in MSD-F1 score. The exception were Russian (ru_gsd) and Finnish (fi_tdt) for which we obtain an error decrease of 34.88% and 46.71% in MSD-accuracy,[13] respectively.

5.4 Multilingual action representations

We take a closer look at action representations projected into a common multilingual space. We analyze the closest neighbours in each language to certain action. Table 11

[13]We noticed that the official MSD-F1 score of the baseline for these treebanks is reported as 0.

Treebank	Baseline				Lem MLE - MBundle			
	LAcc	Lev-Dist	MAcc	M-F1	LAcc	Lev-Dist	MAcc	M-F1
UD_Catalan-AnCora	98.11	0.03	85.77	95.70	83.47	0.26	81.94	86.79
UD_Spanish-GSD	98.42	0.03	81.90	93.95	93.83	0.10	78.44	85.06
UD_Spanish-AnCora	98.44	0.03	84.27	95.30	84.68	0.24	79.66	84.72
UD_French-GSD	98.04	0.04	84.44	94.81	86.85	0.21	78.59	84.51
UD_Hindi-HDTB	98.58	0.02	80.96	94.14	92.92	0.15	69.43	84.38
UD_Latin-Perseus	88.72	0.23	53.23	77.50	56.02	1.14	30.96	32.14
UD_Lithuanian-HSE	84.76	0.30	43.13	67.41	35.82	1.24	21.39	28.57
UD_Cantonese-HK	92.62	0.28	70.15	77.76	98.57	0.01	23.57	25.76
UD_Chinese-CFL	90.72	0.13	74.65	79.91	99.53	0	23.29	24.71
UD_Yoruba-YTB	95.60	0.05	71.20	81.83	96.12	0.04	20.54	17.5
Mean	94.17	0.13	73.16	87.92	74.94	0.62	50.37	58.81
Median	95.92	0.08	76.40	89.46	78.42	0.44	52.77	62.26

Table 10: Performance of system submitted to SIGMORPHON 2019 Shared Task II against the organizer's baseline, for the best 5 and worst 5 treebanks.

presents a summary of the actions queried and their neighbours. Actions are prepended the language they were projected from in square brackets.

First, let us consider actions involving segments known to signal plurality. In general, we observe that the multilingual space successfully captures associations of word forms in plural number and the actions involved in their lemmatization. For the action [es] `del.A_-s`, we note that the closest actions in *es* and *cs* are those involved in the lemmatization of verbs and nouns in plural, whereas *en* actions include the apostrophe from the genitive case indicator *'s*. We observe similar trends for action [es] `subs.A_-s`, even though it is also associated with modality in verbs. We also observe an association with actions involved in lemmatization of verbs in past participle forms in *en*. Similarly, actions of the form [cs] `*.A_-y` are neighbored by non-trivial actions that go beyond adding or deleting a suffix, e.g. diacritic correction in *es* ('botones'→'botn') and '-ves' inflection in *en* ('lives'→'life').

Finally, let us consider the action [es] `del.A_a` involving the segment '-a', known to signal conditionality in verbs in *es*. As expected, the action is neighbored by actions involved in verb lemmatization in *es* and *en*. However, association with the *en* auxiliary 'would' is successfully captured through action `ins._2_-oul`.

5.5 Actions and Morphological Features

We further analyze the associations between individual morphological features and action labels captured by Lem_{MLE}-MBundle. Figure 9 shows the distribution of individual

Query Action	Neighbour Actions	Example (form,lemmata)
[es] del.A_-s		numerosos,numeroso
	[es] del.A_-mo (0.60)	paguemos, pagar
	[es] subs._9_- (0.42)	barcelonesas, barcelons
	[en] islands (0.48)	islands, island
	[en] del.A_-' (0.28)	company's,company
	[cs] del._5_- (0.61)	kopcch, kopec
	[cs] pjmy (0.58)	Pjmy, pjem
[es] subs.A_-s		caiga,caerse
	[es] ins.A_-s (0.86)	atrevi,atreverse
	[es] instrucciones (0.83)	instrucciones,instruccin
	[en] del._4_-i (0.49)	monies,money
	[en] del.A_-t (0.47)	kept,keep
	[cs] statisce (0.82)	statisce,stotisc
	[cs] subs._5_- (0.77)	neptel,neptel
[cs] del.A_-y		zitky,zitek
	[es] autos (0.82)	autos,auto
	[es] subs._4_- (0.71)	comunes,comn
	[en] aspects (0.78)	aspects,aspect
	[en] subs._3_-f (0.75)	lives,life
	[cs] ins.A_-a (0.80)	korun,koruna
	[cs] subs.A_-a (0.76)	ubytovny,ubytovna
[cs] ins.A_-y		ech,echy
	[es] subs._4_- (0.87)	botones,botn
	[es] subs._5_- (0.84)	alemanes,alemn
	[en] waning (0.86)	waning,wane
	[en] subs._3_-f (0.80)	lives,life
	[cs] del._5_-me (0.95)	reimem,reim
	[cs] subs.A_-um (0.94)	masmdich,masmdium
[cs] subs.A_-y		Roztokch,Roztoky
	[es] ins._4_-ec (0.90)	ofrecida,ofrececido
	[es] del.A_-sim (0.75)	sencillsima,sencillo
	[en] replacing (0.90)	replacing,replace
	[en] subs._4_-c (0.72)	taught,teach
	[cs] subs.A_-k (0.96)	vt,velk
	[cs] trsp.A_-ve (0.96)	lhve,lhev
[es] del.A_-a		preguntara,preguntar
	[es] trsp.A_-re (0.90)	habremos,haber
	[es] subs.A_-ir (0.86)	venga,venir
	[en] subs.A_-y (0.85)	said,say
	[en] ins._2_-oul (0.82)	'd,would
	[cs] subs._7_- (0.85)	transfuzi,transfze
	[cs] subs._9_-sk (0.82)	francouzt,francouzsk

Table 11: Neighbour actions (based on cosine similarity) in the multilingual representation space of actions. Language the action was projected from is indicated in square brackets. Cosine distance from query action is indicated in parenthesis.

morphological features over action labels, as defined in Eq.18 for *cs*. Every row represents how likely a fine-grained feature label is to co-occur with an action performed during lemmatization of a token. On the left, we have co-occurrence distributions of gold actions and gold feature labels. On the right, we have co-occurrence distributions of predicted actions and predicted feature labels. For ease of visualization, we only plot the 50 most frequent action labels and features in the development set. We can observe the lemmatizer and tagger succeed in fitting the gold distribution. This is to be expected since the distribution in Eq.18 depends on $P(F^i|x_{1:i})$ and $P(a_j|a_{1:j})$, which are directly optimized by our models. We provide similar plots for *es*, *en*, *tr*, *ar*, and *de* in Appendix A.2.

This analysis also sheds light on which actions and morphological features the model learns to associate. For example, action `del-A_-y` is strongly associated with features PL, N, and MASC, in accordance with the suffix *y* being a plural marker. Another notable example is that of the prefix *ne* which negates a verb. We observe that action `del-_A-ne` is strongly associated with feature V. We also observe ubiquitous features such as POS (positive polarity), which shows an annotation preference unless the bound morpheme of negation is observed (*ne*).

5.6 Limitations

5.6.1 Fixed gold action sequences

Obtaining gold action sequences as a previous, independent step presents a drawback, as pointed out by Makarov and Clematide [2018b]. The optimal action sequence obtained for certain word-lemma pair might not be unique. Hence, if the lemmatizer predicts an alternative valid action sequence, the loss function would still penalize it during training. Given that we consider only one optimal sequence per word-lemma pair, our model cannot take advantage of all the possible valid alternative gold sequences.

5.6.2 Monotonic correspondence assumption

Previous work on neural transducers for morphology tasks Aharoni and Goldberg [2017], Makarov and Clematide [2018c,b] rely on the fact that an almost monotonic alignment of input and output characters exists. This assumption also includes that both words and

lemmas are presented in the same writing system (*same-script condition*), if no off-the-shelf character mapper is used. Our action sequencer relies on the same-script condition in order to not produce too long sequences and in turn, our lemmatizer relies on it to learn meaningful sequences.

During submission to the SIGMORPHON Shared Task, however, we identified a couple of treebanks that violate this condition. In the first one, Arabic-PUD (*ar_pud*), the lemmas are romanized, i.e. presented in Latin rather than Arabic script. For the second one, Akkadian-PISANDUB *(akk_pisandub),* different writing systems (ideographic vs. syllabic) are encoded in the forms but are not preserved in the lemmas. This encoding includes extra symbols such as hyphens and square brackets as well as capitalization of continuous segments. This kind of mismatch between word forms and lemmas forces our lemmatizer to learn action sequences that transform one character at a time, leading to poor performance given our architecture (16.75% and 14.36% on lemmata accuracy for *ar_pud* and *akk_pisandub,* respectively).

5.6.3 Bias towards copying word forms

Languages with little to no morphology such as Chinese or Vietnamese will bias a transducer towards copying the whole input to the output, as pointed out by Makarov and Clematide [2018c]. Our proposed lemmatizers exhibit the same kind of bias, obtaining up to 99.53% of lemmata accuracy for Chinese-CFL and Levenshtein distance of 0.0 in test set and 100% and 0.0 in the development set (see results in Table 12 of Appendix A.1). Other languages benefit from this bias also, as can be observed in Figure 10. We note that, in average, the lemmatizer predicts no more than 3 actions before halting.

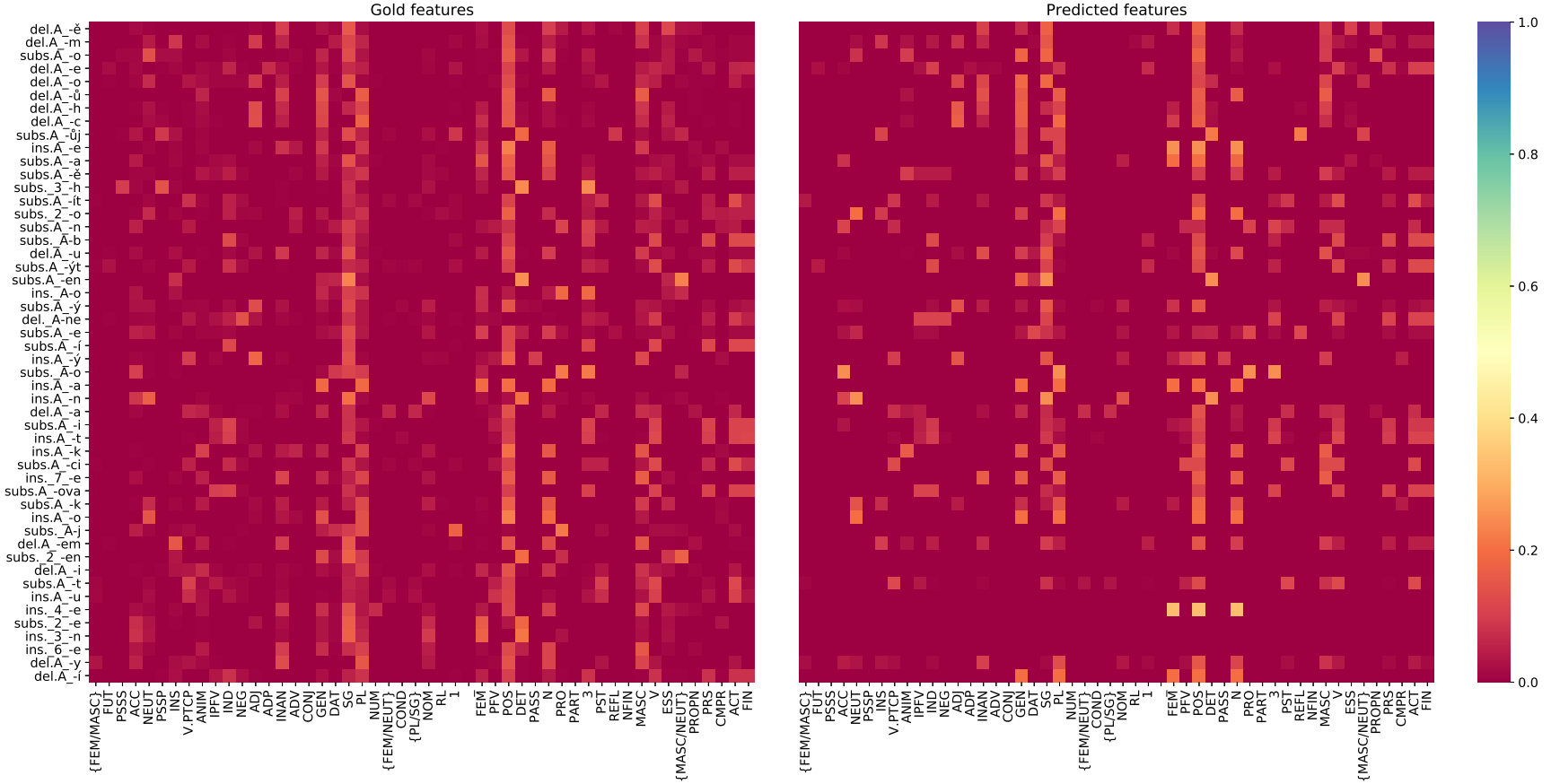

Figure 9: Probability distribution of gold and predicted morphological features given a certain action label, for the Czech-PDT treebank (*cs_pdt*). For ease of visualization, we only plot the 20 most frequent action labels and the 30 most frequent features in the development set.

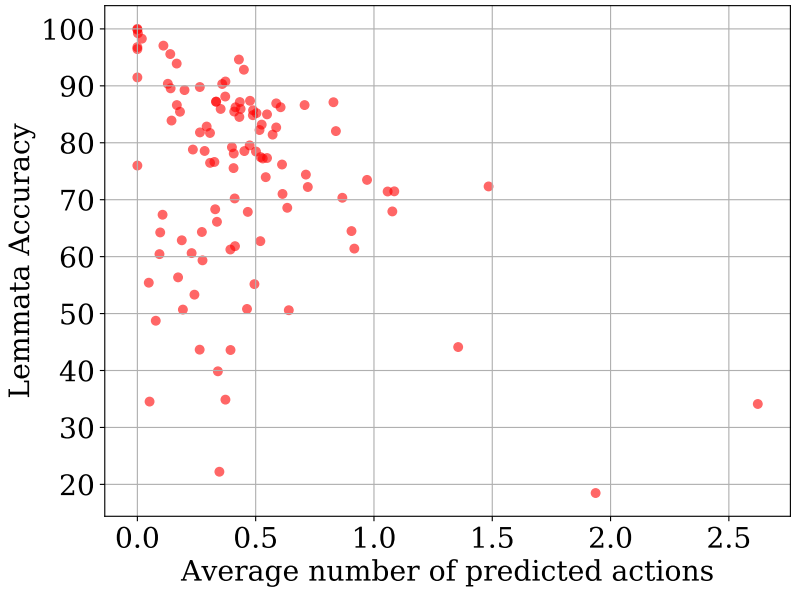

Figure 10: Average number of predicted actions over development set, not including the STOP operation, one data point per treebank.

6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we proposed a lemmatization strategy based on word formation operations derived from extended edit-distance operations that operate at the word level instead of at the character level. These operations are merged using a BPE-inspired algorithm in order to encode segment (e.g. prefix, suffix) information in addition to the action to perform. We find that these operations highly resemble morphological processes, improving prediction interpretability significantly.

For learning word-level actions, we explore maximum likelihood estimate (MLE) and minimum risk training (MRT) as parameter optimization strategies. Our experiments suggest that MRT struggles to further improve over a MLE baseline when the action space is large, e.g. action spaces of highly inflective languages. The harm in performance can be mitigated and even reverted if enough inflections are attested in the training data, as suggested by our results for Czech.

We further analyze what kind of morphological phenomena is captured by our models. First, we analyze a monolingual scenario by observing the co-occurrence of predicted edit actions and predicted morphological features. Our results suggest that our models are better at learning morphological phenomena overmarked through affixation (prefixation and suffixation) and subtraction processes, in comparison to phenomena signaled lexically or by templates. Second, we analyze a multi-lingual learning scenario in which the edit action representations of all languages are projected into a common space. We query action labels involving affixation and subtraction processes known to signal specific phenomena in a language, e.g. Plurality, and inspect whether action labels that signal the same phenomena in other languages can be retrieved. We find that the model learns to group together action labels signaling the same phenomena in several languages, irrespective of the language-specific morphological process that may be involved.

In regards to the task of morphological tagging, we presented several architectures that effectively incorporate sentential context by encoding operation representations hierarchically. Our experiments suggest that predicting MSD labels as bundles yields better

results for all languages except English and Turkish, in comparison with predicting a sequence of individual fine-grained feature labels. These results could be explained by a better annotation quality in terms of consistency on the order of individual morphological feature labels. For English, this results might also be due to the relatively small MSD tagset, the smallest among tagsets of all other analyzed languages, making the task easier.

In addition, we find that using actions projected into the representation space of a highly inflective and morphologically expressive language (in our case, Spanish) further improves tagging performance significantly for all languages.

6.2 Future Work

A potential future research avenue is to tackle the dependency of our approach over fixed gold action sequences. One possible path consists of including the derivation of all possible action sequences as part of the learning pipeline. Makarov and Clematide [2018b] formulated the problem as an imitation learning instance and obtained a completely end-to-end training pipeline.

Another attractive potential future path is to tackle the sparsity of the edit action space, especially action labels with inner position ('_i_' symbol). In this case, the combination of transduction at different levels of granularity, i.e. word level and character level, seems like an attractive strategy. The model would be able to learn alternations between word level actions, suitable for easily identifiable operations or complete lexical substitutions, and character level actions, more suitable for inner-word, one-character operations.

Bibliography

Roee Aharoni and Yoav Goldberg. Morphological inflection generation with hard monotonic attention. *arXiv preprint arXiv:1611.01487*, 2016.

Roee Aharoni and Yoav Goldberg. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1183. URL `https://www.aclweb.org/anthology/P17-1183`.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005. Association for Computational Linguistics, 2012.

Aditi Chaudhary Elizabeth Salesky Gayatri Bhat, David R Mortensen Jaime G Carbonell, and Yulia Tsvetkov. Cmu-01 at the sigmorphon 2019 shared task on crosslinguality and context in morphology. *SIGMORPHON 2019*, page 57, 2019.

Ronald Cardenas, Claudia Borg, and Daniel Zeman. CUNI–malta system at SIGMORPHON 2019 shared task on morphological analysis and lemmatization in context: Operation-based word formation. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 104–112, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4213. URL `https://www.aclweb.org/anthology/W19-4213`.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. The SIGMORPHON 2016 shared task—morphological reinflection. In

Proceedings of the 2016 Meeting of SIGMORPHON, Berlin, Germany, August 2016. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, et al. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. *arXiv preprint arXiv:1706.09031*, 2017.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, et al. The conll–sigmorphon 2018 shared task: Universal morphological reinflection. *arXiv preprint arXiv:1810.07125*, 2018.

William Croft. Parts of speech as language universals and as language-particular categories. *Empirical Approaches to Language Typology*, pages 65–102, 2000.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*, pages 241–249. Association for Computational Linguistics, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Stefan Daniel Dumitrescu and Tiberiu Boros. Attention-free encoder decoder for morphological processing. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 64–68, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-3007. URL `https://www.aclweb.org/anthology/K18-3007`.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. *Ethnologue: Languages of the world*. SIL international, 22nd edition, 2019.

Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75, 1986.

Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

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

Alex Graves. Sequence transduction with recurrent neural networks. In *Proceedings of the Representation Learning Worksop, ICML 2012*, 2012.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

Harald Hammarstrm, Robert Forkel, and Martin Haspelmath. glottolog/glottolog: Glottolog database 4.0, June 2019. URL `https://doi.org/10.5281/zenodo.3260726`.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Rashed Rubby Riyadh, and Grzegorz Kondrak. Cognate projection for low-resource inflection generation. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 6–11, Florence, Italy, August 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W19-4202`.

Georg Heigold, Guenter Neumann, and Josef van Genabith. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 505–513, 2017.

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

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. UniMorph 2.0: Universal Morphology. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan, May 2018. European Language Resource Association. URL `https://www.aclweb.org/anthology/L18-1293`.

Dan Kondratyuk. Cross-lingual lemmatization and morphology tagging with two-stage multilingual bert fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, 2019.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, 2018.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a. URL `https://openreview.net/forum?id=rkYTTf-AZ`.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018b. URL `https://openreview.net/forum?id=H196sainb`.

Chu-Cheng Lin, Hao Zhu, Matthew R. Gormley, and Jason Eisner. Neural finite-state transducers: Beyond rational relations. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 272–283, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1024.

Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

Andreas Madsack and Robert Weißgraeber. AX semantics' submission to the SIGMORPHON 2019 shared task. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–5, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-4201.

Peter Makarov and Simon Clematide. UZH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75, Brussels, October 2018a. Association for Computational Linguistics. doi: 10.18653/v1/K18-3008. URL https://www.aclweb.org/anthology/K18-3008.

Peter Makarov and Simon Clematide. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, 2018b.

Peter Makarov and Simon Clematide. Neural transition-based string transduction for limited-resource setting in morphology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 83–93, 2018c.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. A simple joint model for improved contextual neural lemmatization. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

P. H. Matthews. *Morphology*. Cambridge University Press., 2 edition, 1991.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium, November 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-6011`.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. The SIGMORPHON 2019 shared task: Crosslinguality and context in morphology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy, Aug 2019. Association for Computational Linguistics.

Mehryar Mohri. Weighted finite-state transducer algorithms. an overview. In *Formal Languages and Applications*, pages 551–563. Springer, 2004.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*, 2018.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal dependencies 1.0, 2015. URL `http://hdl.handle.net/11234/1-1464`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel

Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More,

Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayo Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. Universal dependencies 2.3, 2018. URL `http://hdl.handle.net/11234/1-2895`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė,

Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Andre Kaasen, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya,

Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gert-

jan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Taksum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. Universal dependencies 2.4, 2019. URL `http://hdl.handle.net/11234/1-2988`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ramakanth Pasunuru and Mohit Bansal. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*, 2018.

Hao Peng, Roy Schwartz, Sam Thomson, and Noah A Smith. Rational recurrences. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1214, 2018.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, 2012.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *Proceedings of ICLR 2015*, 2015.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633, 2016.

Fynn Schröder, Marcel Kamlot, Gregor Billing, and Arne Köhn. Finding the way from ä to a: Sub-character morphological inflection for the SIGMORPHON 2018 shared task. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 76–85, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-3009. URL https://www.aclweb.org/anthology/K18-3009.

Roy Schwartz, Sam Thomson, and Noah A. Smith. SoPa: Bridging CNNs, RNNs, and Weighted Finite-State Machines. 2018. ISSN 0099-2240. URL http://arxiv.org/abs/1805.06061.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Uygun Shadikhodjaev and Jae Sung Lee. Cbnu system for sigmorphon 2019 shared task 2: a pipeline model. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 19–24, 2019.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Milan Straka, Jana Straková, and Jan Hajic. Udpipe at sigmorphon 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, 2019.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. Noise-aware character alignment for bootstrapping statistical machine transliteration from bilingual corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 204–209, 2013.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

John Sylak-Glassman. The composition and use of the universal morphological feature schema (unimorph schema). *Johns Hopkins University*, 2016.

Ahmet Üstün, Rob van der Goot, Gosse Bouma, and Gertjan van Noord. Multi-team: A multi-attention, multi-decoder approach to morphological analysis. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 35–49, 2019.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics, 1996.

Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, 2016.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2048–2057. JMLR.org, 2015. URL `http://dl.acm.org/citation.cfm?id=3045118.3045336`.

Lei Yu, Jan Buys, and Phil Blunsom. Online segment to segment neural transduction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1307–1316, 2016.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *LREC*, volume 2008, pages 28–30, 2008.

Chunting Zhou and Graham Neubig. Morphological inflection generation with multi-space variational encoder-decoders. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 58–65, 2017.

Appendices

A.1 Results of Submission to SIGMORPHON 2019 Shared Task II

Treebank	Baseline		Lem_{MLE}	
	LAcc	Lev-Dist	LAcc	Lev-Dist
UD_Afrikaans-AfriBooms	98.41	0.03	90.37	0.18
UD_Akkadian-PISANDUB	66.83	0.87	14.36	4.26
UD_Amharic-ATT	98.68	0.02	100.0	0.00
UD_Ancient_Greek-Perseus	94.44	0.14	69.23	0.96
UD_Ancient_Greek-PROIEL	96.68	0.08	73.11	0.84
UD_Arabic-PADT	94.49	0.16	64.63	1.24
UD_Arabic-PUD	85.24	0.41	16.75	5.37
UD_Armenian-ArmTDP	95.39	0.08	66.57	0.80
UD_Bambara-CRB	87.02	0.27	64.84	0.70
UD_Basque-BDT	96.07	0.09	73.81	0.68
UD_Belarusian-HSE	89.70	0.17	59.37	0.80
UD_Breton-KEB	93.53	0.16	64.98	1.00
UD_Bulgarian-BTB	97.37	0.07	81.84	0.52
UD_Buryat-BDT	88.56	0.27	58.65	1.09
UD_Cantonese-HK	91.61	0.28	98.57	0.01
UD_Catalan-AnCora	98.07	0.04	83.47	0.26
UD_Chinese-CFL	93.26	0.10	99.53	0.00
UD_Chinese-GSD	98.44	0.02	99.16	0.01
UD_Coptic-Scriptorium	95.80	0.09	84.71	0.37
UD_Croatian-SET	95.32	0.09	78.59	0.40
UD_Czech-CAC	97.82	0.05	86.25	0.29
UD_Czech-CLTT	98.21	0.04	79.49	0.44
UD_Czech-FicTree	97.66	0.04	85.79	0.28
UD_Czech-PDT	96.06	0.06	85.72	0.26
UD_Czech-PUD	93.58	0.10	49.43	0.96
UD_Danish-DDT	96.16	0.06	80.35	0.33
UD_Dutch-Alpino	97.35	0.05	87.11	0.23
UD_Dutch-LassySmall	96.63	0.06	78.03	0.37
UD_English-EWT	97.68	0.12	88.67	0.16
UD_English-GUM	97.41	0.05	84.96	0.25
UD_English-LinES	98.00	0.04	89.71	0.18
UD_English-ParTUT	97.66	0.04	85.61	0.22
UD_English-PUD	95.29	0.07	81.56	0.28
UD_Estonian-EDT	94.84	0.11	75.48	0.54
UD_Faroese-OFT	88.86	0.2	55.72	0.95
UD_Finnish-FTB	94.88	0.11	70.63	0.80
UD_Finnish-PUD	88.27	0.24	40.71	1.59
UD_Finnish-TDT	95.53	0.10	67.16	0.88
UD_French-GSD	97.97	0.04	86.85	0.21
UD_French-ParTUT	95.69	0.07	89.83	0.20
UD_French-Sequoia	97.67	0.05	86.07	0.25
UD_French-Spoken	97.98	0.04	87.79	0.25
UD_Galician-CTG	98.22	0.04	90.07	0.16
UD_Galician-TreeGal	96.18	0.06	83.24	0.29
UD_German-GSD	96.26	0.08	68.32	0.45
UD_Gothic-PROIEL	96.53	0.07	71.96	0.73
UD_Greek-GDT	96.76	0.07	71.25	0.71
UD_Hebrew-HTB	96.72	0.06	85.71	0.25
UD_Hindi-HDTB	98.6	0.02	92.92	0.15
UD_Hungarian-Szeged	95.17	0.10	66.54	0.83
UD_Indonesian-GSD	99.37	0.01	93.99	0.10
UD_Irish-IDT	91.69	0.18	76.14	0.56
UD_Italian-ISDT	97.38	0.05	85.55	0.26
UD_Italian-ParTUT	96.84	0.08	84.57	0.31
UD_Italian-PoSTWITA	95.6	0.11	78.53	0.42
UD_Italian-PUD	95.59	0.08	77.53	0.44
UD_Japanese-GSD	97.71	0.04	93.64	0.08
UD_Japanese-Modern	94.20	0.07	91.14	0.11
UD_Japanese-PUD	95.75	0.07	94.58	0.07

UD_Komi_Zyrian-IKDP	78.91	0.38	68.75	0.67
UD_Komi_Zyrian-Lattice	82.97	0.34	63.74	0.89
UD_Korean-GSD	92.25	0.18	59.68	0.87
UD_Korean-Kaist	94.61	0.09	73.86	0.56
UD_Korean-PUD	96.41	0.06	27.62	1.56
UD_Kurmanji-MG	92.29	0.39	64.96	0.73
UD_Latin-ITTB	98.17	0.04	87.54	0.34
UD_Latin-Perseus	89.54	0.21	56.02	1.14
UD_Latin-PROIEL	96.41	0.08	72.89	0.77
UD_Latvian-LVTB	95.59	0.07	77.85	0.41
UD_Lithuanian-HSE	86.42	0.25	35.82	1.24
UD_Marathi-UFAL	75.61	0.86	47.97	1.34
UD_Naija-NSC	99.33	0.01	97.24	0.03
UD_North_Sami-Giella	93.04	0.14	60.55	1.05
UD_Norwegian-Bokmaal	98.00	0.03	88.58	0.16
UD_Norwegian-Nynorsk	97.85	0.04	87.80	0.18
UD_Norwegian-NynorskLIA	96.66	0.08	87.28	0.24
UD_Old_Church_Slavonic-PROIEL	96.38	0.08	72.89	0.8
UD_Persian-Seraji	96.08	0.19	84.72	0.59
UD_Polish-LFG	95.82	0.08	78.42	0.45
UD_Polish-SZ	95.18	0.08	70.88	0.57
UD_Portuguese-Bosque	97.08	0.05	79.31	0.33
UD_Portuguese-GSD	93.70	0.18	64.25	1.04
UD_Romanian-Nonstandard	95.86	0.08	82.34	0.38
UD_Romanian-RRT	96.94	0.05	83.48	0.32
UD_Russian-GSD	95.67	0.07	75.81	0.47
UD_Russian-PUD	91.85	0.18	51.66	0.89
UD_Russian-SynTagRus	95.92	0.08	85.40	0.3
UD_Russian-Taiga	89.86	0.21	62.01	0.83
UD_Sanskrit-UFAL	64.32	0.85	27.64	1.93
UD_Serbian-SET	96.72	0.06	75.02	0.47
UD_Slovak-SNK	96.14	0.06	77.90	0.42
UD_Slovenian-SSJ	96.43	0.06	79.50	0.39
UD_Slovenian-SST	94.06	0.12	74.70	0.51
UD_Spanish-AnCora	98.54	0.03	84.68	0.24
UD_Spanish-GSD	98.42	0.03	93.83	0.10
UD_Swedish-LinES	95.85	0.08	82.67	0.32
UD_Swedish-PUD	93.12	0.10	65.57	0.62
UD_Swedish-Talbanken	97.23	0.05	86.72	0.24
UD_Tagalog-TRG	78.38	0.49	78.38	0.73
UD_Tamil-TTB	93.86	0.14	52.68	1.49
UD_Turkish-IMST	96.41	0.08	64.32	1.29
UD_Turkish-PUD	86.02	0.34	47.13	1.75
UD_Ukrainian-IU	95.53	0.10	75.85	0.45
UD_Upper_Sorbian-UFAL	91.69	0.12	57.05	0.88
UD_Urdu-UDTB	96.19	0.07	86.51	0.22
UD_Vietnamese-VTB	99.79	0.02	92.41	0.11
UD_Yoruba-YTB	98.84	0.01	96.12	0.04
Mean	94.17	0.13	74.95	0.62
Median	95.92	0.08	78.42	0.44

Table 12: Official results over the test set of system CHARLES-MALTA-01 (Lem_{MLE}) submitted to Task II - *Lemmatization in Context* of the SIGMORPHON 2019 Shared Task. LAcc: lemmata accuracy; Lev-Dist: Levenshtein distance.

Treebank	Baseline		MBundle	
	MAcc	M-F1	MAcc	M-F1
UD_Afrikaans-AfriBooms	84.90	92.87	59.40	60.00
UD_Akkadian-PISANDUB	78.22	80.41	38.12	39.19
UD_Amharic-ATT	75.43	87.57	34.78	42.42
UD_Ancient_Greek-Perseus	69.88	88.97	55.27	61.48
UD_Ancient_Greek-PROIEL	84.55	93.55	61.24	73.10
UD_Arabic-PADT	76.78	91.82	62.28	69.81
UD_Arabic-PUD	63.07	86.35	27.68	39.46
UD_Armenian-ArmTDP	64.38	86.74	36.09	48.83
UD_Bambara-CRB	76.99	88.94	52.77	56.43
UD_Basque-BDT	67.76	87.54	54.38	63.73
UD_Belarusian-HSE	54.22	78.80	26.93	36.44
UD_Breton-KEB	76.52	88.34	38.21	44.55
UD_Bulgarian-BTB	79.64	93.85	64.89	72.07
UD_Buryat-BDT	64.23	80.94	35.38	38.08
UD_Cantonese-HK	68.57	76.80	23.57	25.76
UD_Catalan-AnCora	85.57	95.73	81.94	86.79
UD_Chinese-CFL	76.71	82.05	23.29	24.71
UD_Chinese-GSD	75.97	83.79	46.54	42.56
UD_Coptic-Scriptorium	87.73	93.56	55.36	63.44
UD_Croatian-SET	71.42	90.39	57.7	69.55
UD_Czech-CAC	77.26	93.94	67.77	79.82
UD_Czech-CLTT	72.6	92.61	24.39	44.82
UD_Czech-FicTree	68.34	90.32	59.98	71.12
UD_Czech-PDT	76.70	94.23	69.16	80.70
UD_Czech-PUD	60.67	85.73	23.21	42.29
UD_Danish-DDT	77.22	90.19	59.26	65.61
UD_Dutch-Alpino	82.07	91.25	77.44	79.69
UD_Dutch-LassySmall	76.78	87.97	61.19	63.90
UD_English-EWT	80.17	90.91	76.86	81.79
UD_English-GUM	79.57	89.81	58.66	61.62
UD_English-LinES	80.30	90.58	64.76	69.93
UD_English-ParTUT	80.31	89.46	54.79	59.61
UD_English-PUD	77.59	87.7	37.57	44.03
UD_Estonian-EDT	74.03	91.52	65.13	75.58
UD_Faroese-OFT	65.32	85.73	41.31	57.70
UD_Finnish-FTB	72.89	89.08	50.30	61.96
UD_Finnish-PUD	70.07	87.77	24.22	40.57
UD_Finnish-TDT	74.84	90.66	54.71	67.39
UD_French-GSD	84.20	94.63	78.59	84.51
UD_French-ParTUT	81.67	92.19	48.03	63.21
UD_French-Sequoia	81.50	93.04	61.06	72.35
UD_French-Spoken	94.48	94.8	65.94	66.17
UD_Galician-CTG	86.65	91.35	77.52	75.41
UD_Galician-TreeGal	76.40	89.33	38.66	52.78
UD_German-GSD	68.35	88.91	65.81	78.39
UD_Gothic-PROIEL	81.00	90.02	47.87	62.90
UD_Greek-GDT	77.44	93.45	47.58	65.34
UD_Hebrew-HTB	81.15	91.79	65.57	69.71
UD_Hindi-HDTB	80.60	93.92	69.43	84.38
UD_Hungarian-Szeged	65.9	87.62	33.99	46.81
UD_Indonesian-GSD	71.73	86.12	44.67	52.13
UD_Irish-IDT	67.66	81.58	29.47	40.44
UD_Italian-ISDT	83.72	94.46	77.25	82.69
UD_Italian-ParTUT	83.51	93.88	62.01	73.55
UD_Italian-PoSTWITA	70.09	87.98	63.7	70.15
UD_Italian-PUD	80.78	92.24	51.13	64.24
UD_Japanese-GSD	85.47	90.64	81.07	79.27
UD_Japanese-Modern	94.94	95.64	62.96	63.61
UD_Japanese-PUD	84.33	89.64	57.44	55.59
UD_Komi_Zyrian-IKDP	35.94	59.52	24.22	32.21
UD_Komi_Zyrian-Lattice	45.05	74.12	26.92	34.75
UD_Korean-GSD	79.73	85.9	63.67	59.84
UD_Korean-Kaist	84.3	89.45	66.34	62.26
UD_Korean-PUD	76.78	88.15	26.38	42.65
UD_Kurmanji-MG	68.10	86.54	31.45	48.17

UD_Latin-ITTB	77.68	93.12	65.40	73.71
UD_Latin-Perseus	55.06	78.91	30.96	32.14
UD_Latin-PROIEL	82.16	91.42	54.59	67.44
UD_Latvian-LVTB	70.33	89.55	56.80	65.13
UD_Lithuanian-HSE	41.43	67.39	21.39	28.57
UD_Marathi-UFAL	40.11	69.71	30.08	37.13
UD_Naija-NSC	66.42	76.73	44.83	38.18
UD_North_Sami-Giella	66.87	85.45	35.86	46.31
UD_Norwegian-Bokmaal	81.27	93.17	79.04	83.01
UD_Norwegian-Nynorsk	81.75	92.85	77.13	81.82
UD_Norwegian-NynorskLIA	74.20	89.21	40.23	41.25
UD_Old_Church_Slavonic-PROIEL	84.13	91.17	51.44	64.19
UD_Persian-Seraji	86.84	93.76	74.13	76.96
UD_Polish-LFG	65.72	88.73	57.84	66.24
UD_Polish-SZ	63.15	86.24	44.82	54.91
UD_Portuguese-Bosque	78.05	92.36	64.79	72.86
UD_Portuguese-GSD	83.87	91.73	70.59	68.01
UD_Romanian-Nonstandard	74.71	91.7	72.54	79.16
UD_Romanian-RRT	81.62	93.88	74.87	80.18
UD_Russian-GSD	63.37	87.49	46.87	57.30
UD_Russian-PUD	60.68	84.31	23.02	41.97
UD_Russian-SynTagRus	73.64	92.73	73.22	78.53
UD_Russian-Taiga	52.06	76.77	25.61	32.5
UD_Sanskrit-UFAL	29.65	57.8	18.09	44.54
UD_Serbian-SET	77.05	91.75	51.43	64.67
UD_Slovak-SNK	64.04	88.04	48.35	60.90
UD_Slovenian-SSJ	73.82	90.12	51.13	65.00
UD_Slovenian-SST	69.57	82.28	30.82	45.63
UD_Spanish-AnCora	84.35	95.35	79.66	84.72
UD_Spanish-GSD	81.90	93.95	78.44	85.06
UD_Swedish-LinES	76.93	89.99	57.43	66.81
UD_Swedish-PUD	79.97	90.49	22.15	41.72
UD_Swedish-Talbanken	81.37	92.65	63.10	73.05
UD_Tagalog-TRG	67.57	87.07	29.73	41.13
UD_Tamil-TTB	73.33	89.22	23.10	47.54
UD_Turkish-IMST	62.94	86.10	30.82	47.29
UD_Turkish-PUD	66.30	87.62	17.27	44.09
UD_Ukrainian-IU	63.59	86.81	42.99	52.07
UD_Upper_Sorbian-UFAL	57.70	81.04	30.63	33.93
UD_Urdu-UDTB	69.97	89.46	57.83	77.83
UD_Vietnamese-VTB	69.42	78.00	44.8	41.86
UD_Yoruba-YTB	73.26	85.47	20.54	17.50
Mean	73.16	87.92	50.37	58.81
Median	76.40	89.46	52.77	62.26

Table 13: Official results over the test set of system CHARLES-MALTA-01 (MBundle) submitted to Task II - *Morphological Analysis in Context* of the SIGMORPHON 2019 Shared Task. MAcc: MSD 0/1 accuracy; M-F1: MSD F1-score (micro-averaged).

A.2 Actions and Morphological Features

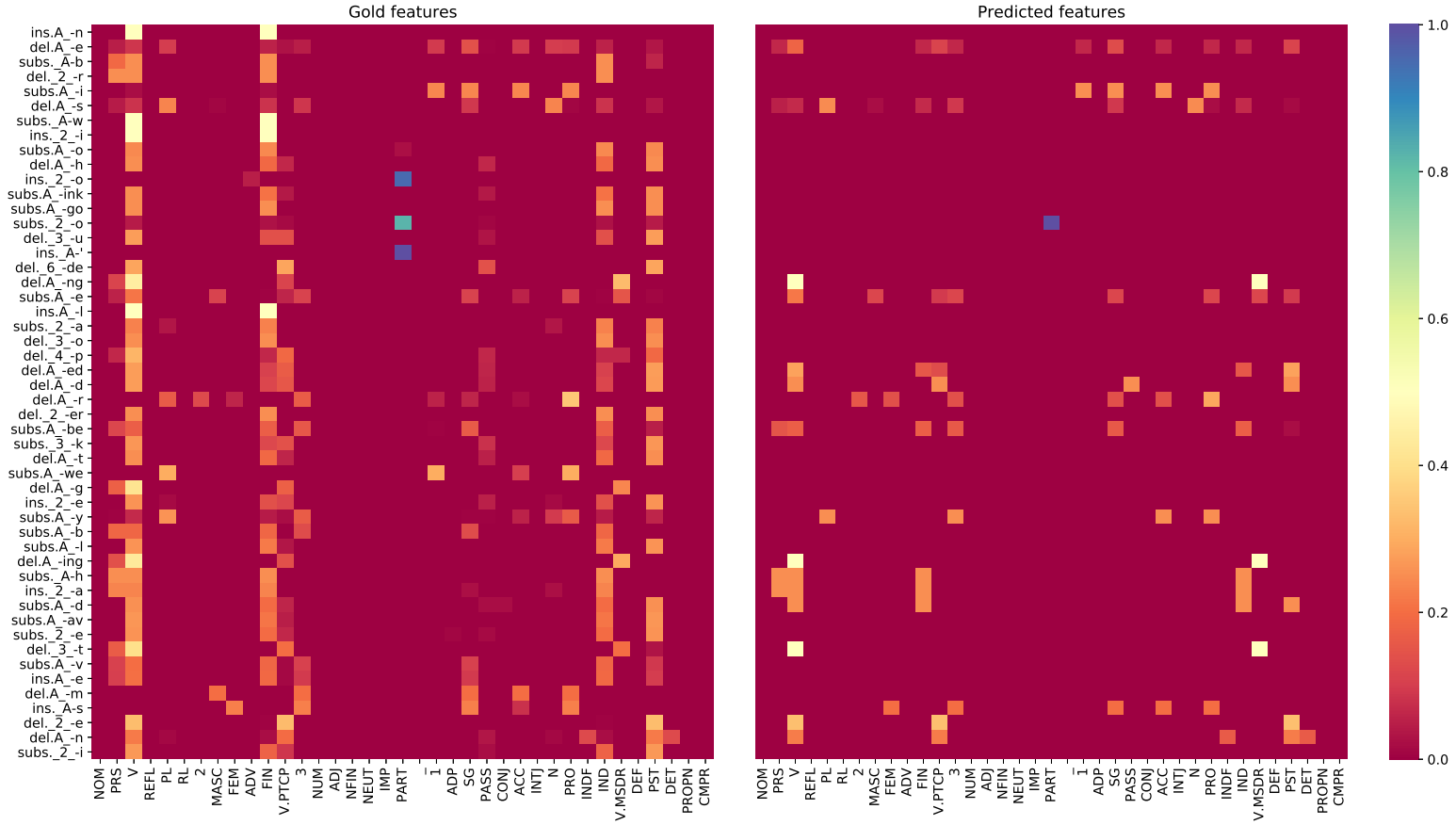

Figure 11: Probability distribution of gold and predicted morphological features given a certain action label, for English (en_ewt).

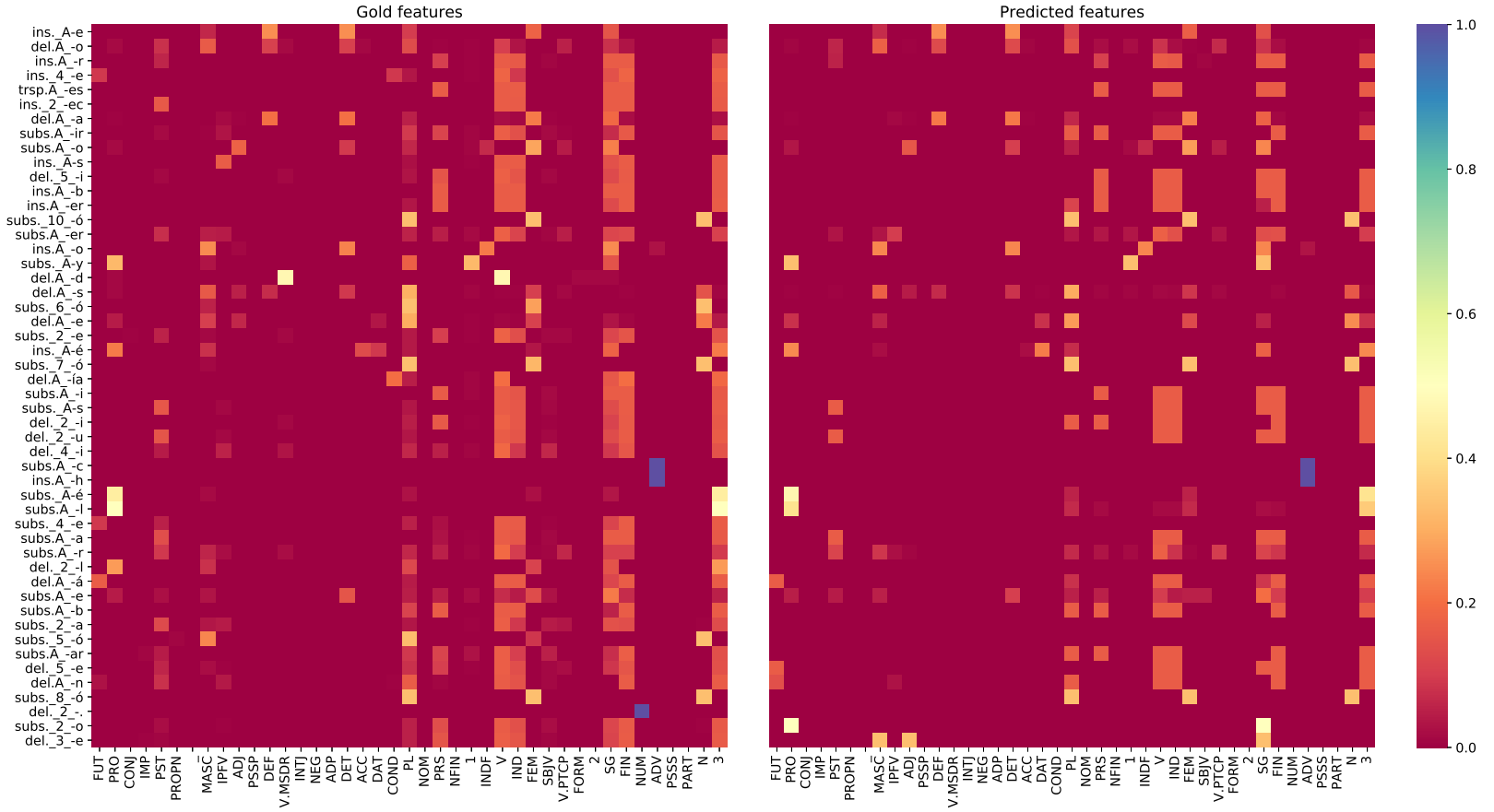

Figure 12: Probability distribution of gold and predicted morphological features given a certain action label, for Spanish (es_ancora).

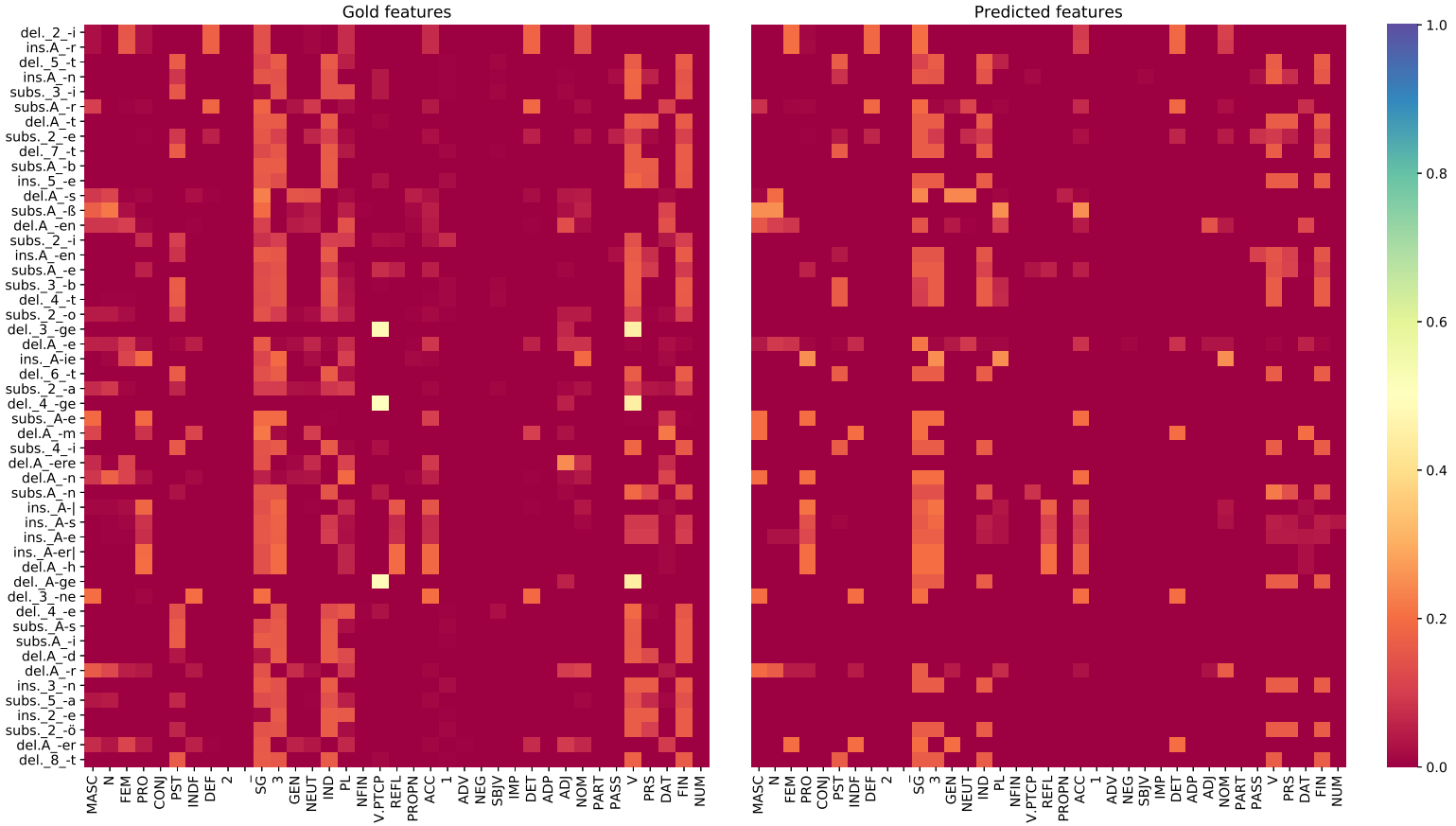

Figure 13: Probability distribution of gold and predicted morphological features given a certain action label, for German (de_gsd).

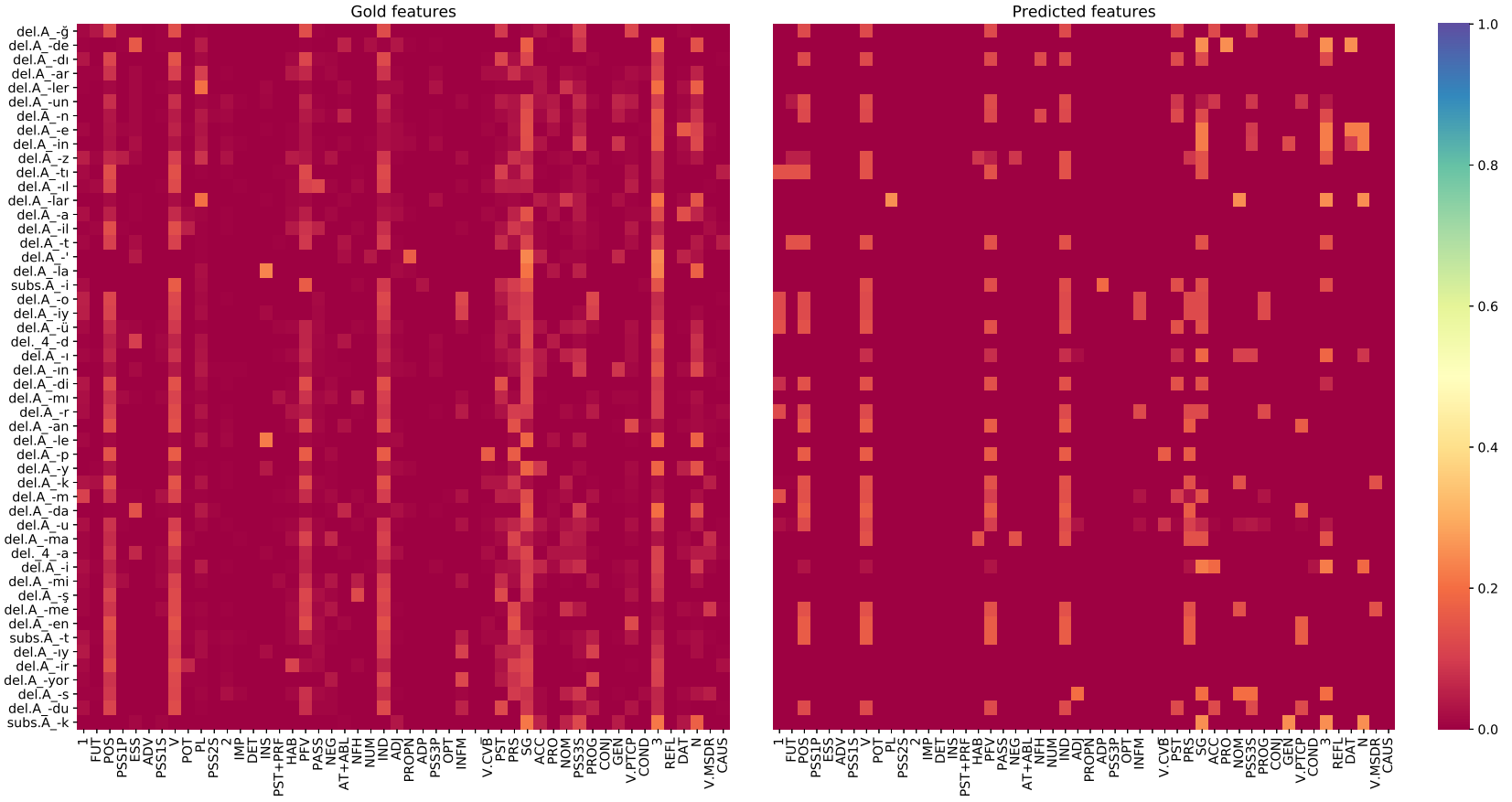

Figure 14: Probability distribution of gold and predicted morphological features given a certain action label, for Turkish (tr_imst).

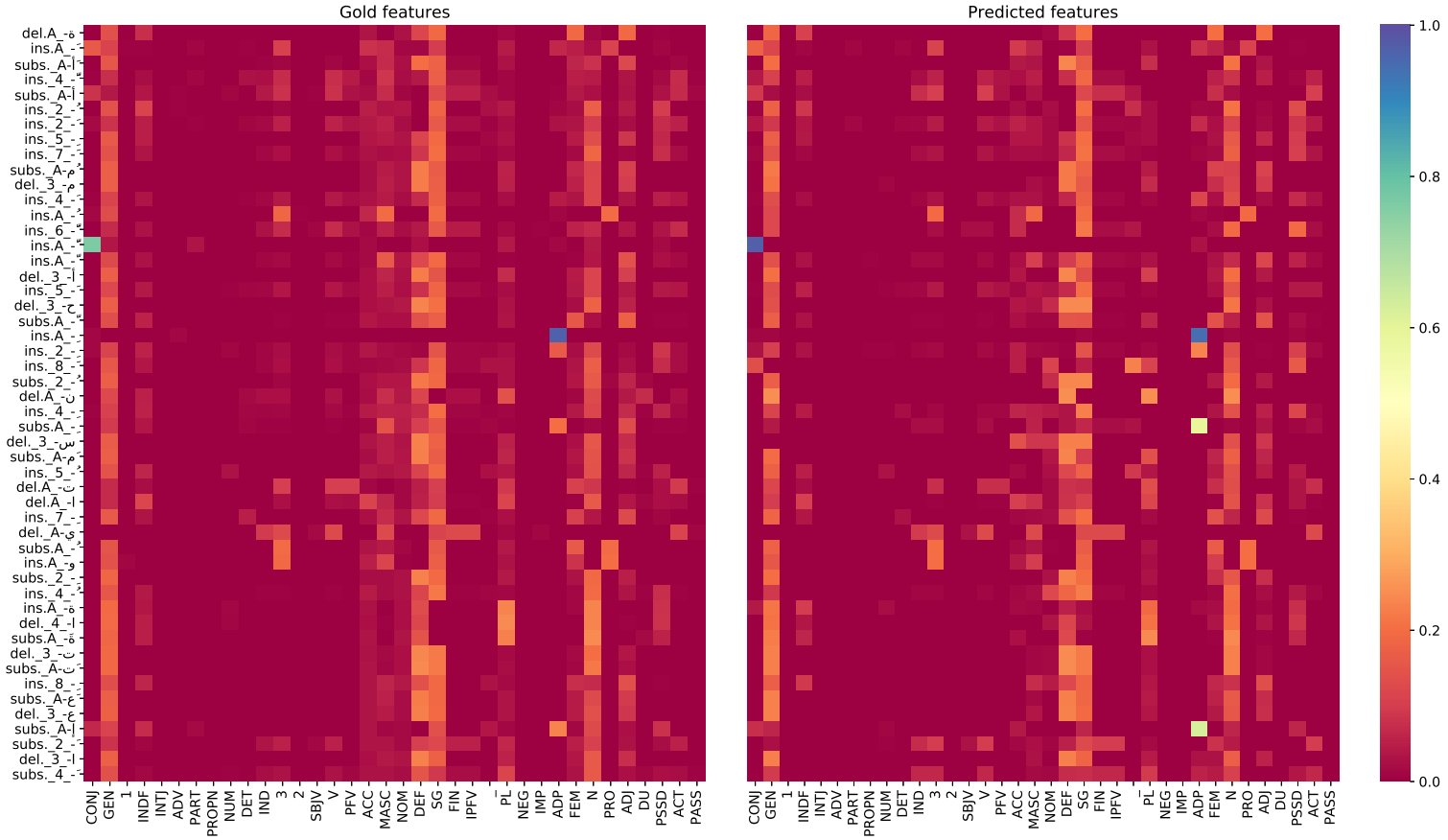

Figure 15: Probability distribution of gold and predicted morphological features given a certain action label, for Arabic (ar_padt).

List of Figures

List of Tables

List of Abbreviations

WFSA	Weighted finite state automata
WFST	Weighted finite state transducer
POS	Part of Speech
MSD	Morpho-syntactic description
RL	Reinforcement Learning
FST	Finite state transducer
UD	Universal Dependencies
UPOS	Part of Speech tagset in Universal Dependencies
UFEAT	Morpho-syntactic description tagset in Universal Dependencies
Seq2Seq	Sequence-to-sequence neural architecture
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long short-term memory
MLE	Maximum likelihood estimate
MRT	Minimum risk training
LAcc	Lemmata accuracy
Lev-Dist	Levenshtein distance
MAcc	MSD 0/1 accuracy
M-F1	micro-average MSD F1-score