Modeling Morphological Typology for Unsupervised Learning of Language Morphology

Hongzhi Xu[1,3], **Jordan Kodner**[2], **Mitch Marcus**[1], **Charles Yang**[2]
[1]CIS Department, University of Pennsylvania, Philadelphia, USA
[2]Linguistics Department, University of Pennsylvania, Philadelphia, USA
[3]ICSA Institute, Shanghai International Studies University, Shanghai, China
hongz.xu@gmail.com, jkodner@sas.upenn.edu
mitch@cis.upenn.edu, charles@ling.upenn.edu

Abstract

This paper describes a language-independent model for fully unsupervised morphological analysis that exploits a universal framework leveraging morphological typology. By modeling morphological processes including suffixation, prefixation, infixation, and full and partial reduplication with constrained stem change rules, our system effectively constrains the search space and offers a wide coverage in terms of morphological typology. The system is tested on nine typologically and genetically diverse languages, and shows superior performance over leading systems. We also investigate the effect of an oracle that provides only a handful of bits per language to signal morphological type.

1 Introduction

Morphological analysis aims to identify languages' word-internal structures. Early approaches to the computational analysis of morphology modeled the structure of each language with hand-built rules, (e.g. Sproat, 1992). Such systems require a significant amount of work from domain experts, and while they tend to be very accurate, they also suffer from low coverage. Supervised and semi-supervised machine learning approaches require expert input and will suffer from out-of-vocabulary problems. This paper focuses primarily on fully unsupervised morphological learning, which offers the most flexibility and can be deployed for new languages with no data annotation.

Concatenation-based morphological learning systems aim to identify morphemes or morpheme boundaries within words (Virpioja et al., 2013; Goldwater and Johnson, 2004; Creutz and Lagus, 2005, 2007; Lignos, 2010; Poon et al., 2009; Snyder and Barzilay, 2008). The Morpho-Challenge tasks[1] provide a set of morphologically annotated data for testing concatenation. However, systems designed directly for identifying morpheme boundaries are limited in that non-linear structures such as infixation cannot be well captured.

Another approach exploits morphological relations between word pairs. Related words form morphological chains through processes of derivation. There are many such processes including affixation at the edges or middle of a word, reduplication, stem transformations, and so on. Of these, only edge-affixation is available to concatenation-based models, so leveraging derivation directly allows for wider cross-linguistic coverage (Schone and Jurafsky, 2001; Narasimhan et al., 2015; Soricut and Och, 2015; Luo et al., 2017; Xu et al., 2018).

A more holistic line of work builds learning on the concept of morphological paradigms (Parkes et al., 1998; Goldsmith, 2001; Chan, 2006; Xu et al., 2018). Paradigms can be defined as sets of morphological processes applicable to homogeneous groups of words. For example, the paradigm (*NULL, -er, -est, -ly*) in English can be applied to adjectives (e.g., *high, higher, highest, highly*), while (*NULL, -ing, -ed, -s, -er*) is defined over verbs (e.g, *walk, walking, walked, walks, walker*). Paradigms have several merits. First, they provide a principled strategy for tackling the data sparsity problem. In morphologically rich languages, a single word can derive hundreds of forms most of which will be unattested in real data. This can be addressed by taking paradigms into account because if a word appears in part of the paradigm, it likely can appear in the rest too. The recent SIGMORPHON shared tasks in paradigm filling are along this line (Cotterell et al., 2016, 2017, 2018). Second, paradigms can be used to identify spurious morphological analyses. For example, the words *within, without, wither* might be analyzed as applying suffixes *-in, -out, -er* to the word *with*, however, the paradigm (*-in, -out, -er*) is not reliable since it only applies to

[1]http://morpho.aalto.fi/events/morphochallenge/

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6672–6681
July 5 - 10, 2020.

one single word, i.e. *with*.

One thread common in previous work is the lack of consideration for characteristics of language-specific morphological typology. In this paper, we propose a new framework that incorporates typological awareness by explicitly modeling different morphological patterns including suffixation, prefixation, infixation, and reduplication. These patterns have covered most common morphological processes of the languages in the world, with the exception of templatic morphology which is not represented in the LDC-provided test sets. By building such universal linguistic knowledge, the model will benefit from both constraining the search space (without generating a large amount of spurious analyses) and providing a wider coverage especially for the non-linear morphological structures.

2 Related Work

The Morpho-Challenge tasks held between 2005 and 2010 motivated a large amount of work on unsupervised morphology learning including the Morfessor family of models. The Morfessor baseline system (Creutz and Lagus, 2002; Virpioja et al., 2013), an MDL model, is one of the most popular unsupervised systems for automatic morphological segmentation. Creutz and Lagus (2005, 2007) extend the model with the maximum a posteriori (MAP) on both observed data and the model. These systems only require word lists as input, which is an advantage for low-resource languages where there is no large corpus for training complex models.

Various work has explored the idea of paradigms. Parkes et al. (1998) try to learn inflectional paradigms on English verbs, Goldsmith (2001, 2006) exploits the MDL principle to learn paradigms (referred to as *signatures*) with a greedy search strategy, and Dreyer and Eisner (2011) adopt a semi-supervised log-linear model to identify paradigms, which requires a number of seed paradigms for training. However, in morphologically rich languages such as Turkish where a single paradigm can be extremely large, this method requires considerable human annotation effort. Ahlberg et al. (2014) use a semi-supervised approach to learn abstract paradigms from a given inflection table. However, the task is different from what we discuss here, which discovers inflection tables as an intermediate step. Xu et al. (2018) create paradigms from the results of a probabilistic model and use the reliable paradigms to prune unreliable ones and achieve promising results. Xu et al. (2018)'s model only deals with suffixation. The framework that we develop in this paper is most directly inspired by Xu et al. (2018).

Schone and Jurafsky (2001) use semantic information to identify real morphological pairs from a set of orthographically similar word pairs. Similarly, Soricut and Och (2015) use orthographic information to generate candidate morphological rules, e.g., *prefix* : $: *in*, and then use word embeddings to evaluate the qualities of the rules. Narasimhan et al. (2015) create *morphological chains*, e.g., *(play, playful, playfully)*, using both orthographic information and distributional semantics by maximizing the likelihood through a log-linear model. One drawback of using distributional information is that it requires large text corpora to train reliable semantic vectors. This is a major hurdle for applying such a system to low-resource languages. Based on the output of Narasimhan et al. (2015)'s model, Luo et al. (2017) adopt integer linear programming (ILP) to find globally optimal paradigms, which they call morphological forests, and achieve improved performance.

3 Morphological Typology

This section surveys the morphological phenomena frequently observed among the world's languages which our system is able to account for.

3.1 Prefixation, Suffixation, and Infixation

Affixation is the appending of a bound morpheme or *affix* onto either end of a word and is the most common kind of morphological operation (Dryer, 2013). Affixes postpended to a word are called *suffixes* such as *-ed*, *-ing*, *-ness*, or *-est* in English, while *prefixes* are prepended such as *pre-* or *un-*, and *infixes* find their way into the middle of a root. Infixes are rarer cross-linguistically, but they do surface around the world, notably in languages like Tagalog (Malayo-Polynesian), ***dulot*** $\sim$ ***d****-in-****ulot*** or ***graduate*** $\sim$ ***gr****-um-****aduate***.

Many languages stack or nest affixes. English derivational morphology does this occasionally as in *anti-dis-****establish****-ment-ari-an-ism* or in Shona (S Bantu) inflectional morphology, for example, *ha-mu-cha-mbo-nyatso-ndi-****rov****-es-i=wo* 'You will not cause me to be beaten' (Mugari, 2013). A given affix may never appear on the edge of a word since it can be obligatorily followed or preceded by more affixes. This can be seen in Bantu verbs which nec-

essarily end with a so-called *final vowel* morpheme (here, *-a*). Most other suffixes have to appear before the final vowel, so they are never themselves suffixes in the string sense. For example, given the Shona *ku-**pig**-a* 'to strike,' one could form *ku-**pig**-an-a* 'to strike one another' or *ku-**pig**-w-a* 'to be stricken' but not **ku-**pig**-w* or **ku-**pig**-an*. We will refer to the disconnect between morphological suffixation and string suffixation as the *final vowel problem*.

3.2 Reduplication and Partial Reduplication

Reduplication, the doubling of all or a part of a word, is productive in many languages, especially outside modern Europe (Rubino, 2013). Full reduplication can indicate plural number, repeated actions, or progressive aspect in Austronesian languages such as Indonesian and Tagalog. In Indonesian, sometimes a whole word including its affixes is reduplicated (***bangun**-an-**bangun**-an*), while other times it is only the root (***deg**-**deg**-an* or *ber-**bondong**-**bondong***). Partial reduplication is exemplified in Pangasinan, an Austronesian relative of Tagalog, which has more productive partial reduplication for plurals. It can surface on the left (***plato*** $\sim$ *pa-**pláto***), or it may be infixed (***amigo*** $\sim$ ***ami**-mí-**go***) (Rubino, 2001).

3.3 Stem Changes

Some morphology is expressed through *stem changes* rather than string concatenation. English often expresses past tense, past participles, and plurals with changes to stem vowels, sometimes in conjunction with affixation (***sing*** $\sim$ ***sang*** $\sim$ ***sung***, ***freeze*** $\sim$ ***froze*** $\sim$ ***froz**-en*, and ***goose*** $\sim$ ***geese***). Consonants can alternate as well, for example in Finnish ***luku*** $\sim$ ***luvu**-t* and ***etsi**-**nt**-ä* $\sim$ ***etsi**-**nn**-ät*. Some changes are *morphophonological* because they are related to the phonology of the language and thus are somewhat predictable. For example, the Latin root ***scrib*** becomes ***scrip**-t-us* in the past participle because /b/ is devoiced before /t/. These contrast with alternations like ***goose*** $\sim$ ***geese*** which are arbitrary – there is no ***moose*** $\sim$ ****meese***.

Vowel harmony is a kind of pervasive global morphophonological pattern which forces vowels in a word to share certain features. In the simplest case, this often results in affix *allomorphy* where each affix has alternate forms that agree with the features in the root or the root must agree with the affixes. Finnish presents a classic example of front-back vowel harmony: a word may contain front vowels (*ä*, *ö*, *ÿ*) or back vowels (*a*, *o*, *u*) but not both. Suffixes have front and back allomorphs in order to agree with the stem. For example, contrast the front-containing suffixes after front-containing root ***liity**-**nt**-öjä* with the same suffixes after a back-containing root ***liiku**-**nt**-oja*.

4 Modeling Morphological Processes

In this section, we describe our framework for modeling language morphologies, including prefixation, suffixation, infixation, full and partial reduplication. We also model stem changes that typically occur at word boundaries except for vowel changes.

4.1 Morphology as Lexical Pairs

Many theories of morphology such as paradigm-based morphology, e.g. Paradigm Function Morphology (Stump, 2001), cast morphology as a relation between word pairs. We adopt this perspective as the basis of our framework, except that we do not differentiate derivational morphology from inflection. In detail, the framework assumes morphology to be an operation that is applied to a word (root) to form another word and effects a change in meaning along some dimension, e.g., adding information such as case, number, gender, tense, or aspect. We denote such a morphological process with a function f. The function takes a root word r as input and forms a new word w, i.e. $f(r) = w$. Thus the task of morphology learning can be defined as searching for a function f and another word r, given a word w, such that $f(r) = w$.

4.2 Constraining the Search Space with Morphological Typology

Here, we describe how we incorporate prefixation, suffixation, infixation, and full and partial reduplication to constrain the morphological function space. This improves over naive methods focusing on edit distance, which can be used to evaluate how good a morphological function is locally. Globally, a morphological function can be evaluated by observing its overall frequency, namely its corpus productivity in a language. Such a simple system would tend to hallucinate many spurious yet frequent morphological functions, which may not be possible morphologically from a richer linguistic perspective.

Morphological patterns allow us to represent the derivation of complex words from root words. A prefixation pattern can be defined as <*prefix*>_*x*,

where <*prefix*> is a specific prefix in a language, and x stands for the root. For example, the pattern <*un*->_*x* describes how the word *unfold* can be derived from *fold* with a prefix. A suffixation pattern can be defined as *x*_<*suffix*> and an infixation pattern can be defined similarly as *bx*_<*infix*>_*ex*, where *bx* and *ex* are the beginning and ending part of the root word *x* and *x* = *bx* + *ex*.

Reduplication functions can be defined in the same way. A full reduplication pattern is defined as *x_x*. A partial reduplication can be defined as *bx_x* ($bx \neq x$) with the partial copy of x on the left or *x_ex* ($ex \neq x$) with the partial copy on the right. Table 1 shows all the morphological patterns associated with examples from different languages.

Morphological Type	Eg. Func	Eg. words
Prefixation	<*di*->_x	*di-bangun*[2]
Infixation	bx_<-*in*->_ex	*d-in-ulot*[4]
Suffixation	x_<-ɛ>	*kyer*ɛ-ɛ[1]
Full reduplication	x_x	*kyer*ɛ-*kyer*ɛ[1]
Partial reduplication (L)	bx_x	*ka-kain*[4]
Partial reduplication (R)	x_ex	
Final Vowel / Theme V	x-v<*a*>	*pig-a*[3]

Table 1: Morphological operations with example patterns and words in [1] Akan, [2] Indonesian, [3] Swahili, and [4] Tagalog. No right partial reduplication is present in our test set.

4.3 Morphophonological Rules

Here, we define the stem change rules that are motivated by morphophonological observations on languages which we denote with the function g. We extend the capabilities of previous systems (Narasimhan et al., 2015; Xu et al., 2018) and model six transformation rules as follows:

Insertion (INS) of a letter at the end of the root. E.g. the Spanish word *quiera* can be analyzed as (*quer, -a, INS-i*).

Deletion (DEL) of the end letter of the root. E.g. *using* can be analyzed as (*use, -ing, DEL-e*).

Gemination (GEM) of the end letter of the root. E.g. *stopped* can be analyzed as (*stop, -ed, GEM-p*).

Degemination (DEG) of the end letter of the root if it is in a reduplication form. E.g. the Finnish word *katot* can be analyzed as (*katto, -t, DEG-t*).

Substitution (SUB) of the end letter of the root with another. E.g. the word *carries* can be analyzed as (*carry, -es, SUB-y-i*).

VowelChange (VOW) of the right or left most vowel of the root with another. For example, the word *drunken* can be analyzed as (*drink, -en, VOW-i-u*). This feature requires the system to be aware of a global vowel inventory.

4.4 Generating Candidate Morphological Functions

A *morphological function* is defined as two parts: the morphological pattern, and the corresponding stem changes, $f = [$<*stem_change*>, <*morph_pat*>$]$, where <*stem_change*> is first applied to the root, with the output fed into the <*morph_pat*> to generate the derived word. A detailed definition can be denoted as $f(r) = [g(x),$ <*prefix*>_$x](r)$, where r is the root word which can apply this rule to derive another word, and g is a stem change function.

For example, a prefixation function $f(r) = [\$(x),$ <*un*->_$x](r)$ (where $\$(x)$ means no stem change applies) can be applied to the verb *fold* to generate the verb *unfold*. Similarly, a suffixation function $f(r) = [$*SUB-y-i*$(x), x$_<-*ed*>$](r)$ can be applied to the verb *carry* to generate the verb *carri-ed*. We can define an infixation function $f(r) = [(\$(bx), \$(ex)), bx$_<-*um*->_$ex](r)$; when applied to the word *kakain*, it can generate the verb *k-um-akain*. A full reduplication function can be defined as $f(r) = [\$(x), \$(x)), x\_x](r)$; when applied to the word 'kyerɛ', it can generate the verb *kyer*ɛ-*kyer*ɛ. A partial reduplication function $f(r) = [(\$(bx), \$(x)), bx\_x](r)$, when applied to the word *kain*, can generate the verb *ka-kain*.

The central phase of learning involves generating potential morphological functions. During this phase, no stem changes are allowed in order to limit spurious functions. Learning is done by comparing each word pair and postulating a function f that can explain the pair, where the function f is constrained through morphological typology as described in Section 3. For example, given the word pair (*fold*, *unfold*), we can postulate a prefixation function $f(r) = [\$(x),$ <*un*->_$x](r)$; given word pair (*kain*, *kakain*), we can postulate a left partial reduplication function $f(r) = [(\$(bx), \$(x)), bx\_x](r)$.

For affixation, including prefixation, infixation, and suffixation, a set of candidate affixes is needed before generating morphological functions. This can be done by comparing all possible word pairs, a similar method used by previous studies (e.g. Narasimhan et al., 2015; Xu et al., 2018). For prefixes, if $w = s + w'$, where w and w' are both attested words in the word list, then s is a can-

didate prefix. We use the cardinality of the set $\{(w, w') : w = s + w'\}$ to evaluate how good the candidate prefix s is. Similarly, for suffixes, if $w = w' + s$, then s is a candidate suffix. For infixes, if $w = bw' + s + ew'$, where w and $w' = bw' + ew'$ are both attested words in the word list, then s is a candidate infix. Finally, only the top N most frequent candidates for each affix type are selected.

4.5 Searching for Candidate Analyses for Individual Words

After generating all morphological functions reflecting each morphological type, searching for candidate analyses for individual words is conceptually straightforward. For a given word w, we find all possible morphological functions $\{f : f = [g, m]\}$ associated with a root word r, such that $w = f(r)$. For example, the word *reread* can be analyzed as <*re*->_X, bx_<-*re*->_ex, and bx_x.

This is somewhat complicated by the need to find possible morphophonological (stem change) rules on the root words. The basic idea is that when checking a possible prefixation pattern, for example $w = s + w'$, rather than assuming w' is an attested word, we assume that if there is an attested word w'' and a potential stem change rule g, such that $w' = g(w'')$, then <s>_x is a potential prefixation pattern for w. We can easily create an index based on the attested words to accelerate the searching process. Searching for suffixation and infixation can be done is a similar way.

For reduplication, we use a similar strategy. If $w = bw' + w'$, i.e. a word w can be decomposed into another word w' plus a string prefix of w' on the left, then we postulate a partial reduplication pattern for word w, i.e. bx_x. If $w = w' + ew'$, then x_ex can be generated. For example, given that the word *reread* = *re* + *read* and *read* is itself a word, we can hypothesize that the word is bx_x. For full reduplication, if a word $w = w' + w'$, where w' is another word, then a morphological pattern x_x can be generated for w.

For more complicated cases, we extend the search for reduplication of individual words with possible stem change rules. For partial reduplication, if a word $w = s + w'$, and there is a stem change rule g, such that $s = g(bw')$, then we can also postulate a partial reduplication pattern for w, with a stem change rule on bw'. Similarly, if a word $w = s + s'$, and there is a stem change function g and an attested word w' such that $s' = g(w')$ and $s = bw'$, then we can also postulate a partial reduplication pattern for w with a stem change rule on w'. For full reduplication, if a word $w = s + s'$, there are (up to) two stem change functions g and g', and a word w', such that $s = g(w')$ and $s' = g'(w')$, then we can postulate a full reduplication pattern for w.

4.5.1 Further Decreasing the Search Space

A large number of spurious candidate analyses will be generated once we allow stem change rules. However, some candidate analyses can be ruled out given other candidates. For example, the word 'saying' can be analyzed as (*say*, $, x_<-*ing*>), but also as (*says*, DEL-<*s*>, x_<-*ing*>), but the latter one is unnecessary given the former one and a heuristic that says that no stem changes are to be preferred to stem changes. So, to further decrease the search space, we employ a set of heuristics to eliminate some of the candidate analyses before the next step. They follow a principle of parsimony, namely once a simpler analysis is generated, the more complicated ones that are related will be excluded.[2]

5 Disambiguation with a Probabilistic Model

After generating all candidate analyses for a given word, we evaluate how good each candidate is so we can choose the best one as the final analysis. We compute the conditional probability of a candidate analysis $[g, m](r)$ given a word $w = [g, m](r))$, namely $P(r, g, m|w)$. $P(r, g, m|w) = 0$ if $[g, m](r) \neq w$. Otherwise, we use the following formula to calculate this probability.

$$P(r, g, m|w) = \frac{P(r, g, m)}{\sum_{(r', g', m')=w} P(r', g', m')} \quad (1)$$

To compute the probability of a candidate analysis $(w = [g, m](r))$, $P(r, g, m)$, we assume that r, g and m are independent to each other. So,

$$P(r, g, m) = P(r) \times P(g) \times P(m) \quad (2)$$

The probabilities in this model can be estimated using EM initialized by counting all the candidate analyses of all words in the word list and assuming that each candidate has the same probability.

[2]The details will be given in a separate document with the code that will be made publicly available before the conference.

5.1 Solving Oversegmentations with Paradigms

We extend Xu et al. (2018)'s work and use statistically reliable paradigms for filtering unreliable ones. In detail, a paradigm is defined by Xu et al. (2018) upon a set of suffixes. Here, we extend this definition to a mixture of different types of morphological processes, i.e. $M = \{m\}$, that can be applied to the same set of roots $R = \{r\}$ to be in a paradigm. Formally, a paradigm is defined as $p = R \times M$. Finally, the paradigms with at least 2×2 sizes are selected as reliable ones, namely at least two morphological patterns supported by at least two roots. Similar to Xu et al. (2018), stem changes are not part of the paradigm since they are generally independent processes.

After finding possible paradigms, we use the same method for pruning unreliable paradigms. Given an unreliable paradigm $p = R \times M$, the intersection of the morphological pattern set M and the set M_i of each reliable paradigm p_i is computed, i.e. $M'_i = M \cap M_i$, and the one with the best score, e.g. M'_k will be chosen as the pruned result, i.e. $p' = R \times M'_k$. Finally, the score of an intersection M'_i is the sum of the frequencies of all the morphological patterns in the intersection, as shown in equation 3.

$$score(M) = \sum_{m \in M} freq(m) \quad (3)$$

5.2 Generating Morphological Derivations

After the one-step roots of all the words are found, morphological derivations (e.g., *sterile*, *sterilize*, *sterilizing*) are automatically generated iteratively by our system as well as final segmentations (e.g., *steril-iz-ing*). As described in the next section, because evaluation will be based on morpheme boundaries identification, generating such a segmentation is necessary.

6 Experiments

6.1 Settings

We compare our model with Morfessor (Virpioja et al., 2013), the most popular baseline, MorphoChain (MC) (Narasimhan et al., 2015) and its improved version, Morph-Forest (MF) (Luo et al., 2017), and ParaMA (PMA) (Xu et al., 2018). We evaluate the models with segmentation points (boundaries of morphemes), the same metric used by Narasimhan et al. (2015) and Xu et al. (2018). We run our model in two different settings. In the primary experiment, we run it as a fully unsupervised model (FU), assuming all possible typological features. In a secondary experiment, each language's morphological typology is provided by an oracle so that the model can only search relevant patterns per language (U+T). A vowel inventory is also provided so that our system can discover the vowel change rules described in Section 4.3. MC and MF are run in two different configurations, one with semantic vectors (+v) and the other without vectors (more comparable to Morfessor, ParaMA, and our system).

Lang	Train	Test	Corpus	Morphology
Aka	74K	2K	3M	pref, suf, red
Hin	487K	2K	28M	pref, suf, red
Hun	4,390K	2K	574M	pref, suf
Ind	525K	2K	19M	pref, suf, inf, red
Rus	1,485K	2K	1,068M	pref, suf
Spa	564K	2K	24M	pref, suf
Swa	224K	2K	4M	pref, suf, red, fv
Tag	13K	2K	5M	pref, suf, inf, lred, red
Tam	2,363K	2K	47M	pref, suf, red

Table 2: Number of word types for training and testing, corpus size for training word vectors (only for MorphoChain and Morph-Forest systems), and the morphological features (pref: prefixation; suf: suffixation; inf: infixation; red: full reduplication; lred: left reduplication; fv: final vowel) for each language.

We conduct the experiments with a data set containing 9 languages from diverse language families (Mott et al., 2020). The details of the data sets including the typological features for each language and the size of corpus that is used for training word vectors are shown in Table 2. The word lists used for training are extracted from the language pack created under the DARPA LORELEI (LOw REsource Languages and Emergent Incidents) program. The gold standard data, soon to be released by LDC, is annotated only with morpheme segmentations, and no data annotation was used in training. The languages with non-Latin scripts were romanized with the tools provided in the package.

6.2 Experimental Results and Analyses

Results are presented in Figure 1. The details are shown in Table 3 and Table 4. Both our unsupervised model (FU) and model with given typology (U+T) achieve higher average F1 than previous work by a large margin, the highest on five of nine languages, and competitive results overall on the other four. Of the two systems, the typology feature oracle provided only slightly better average performance than fully unsupervised. As expected,

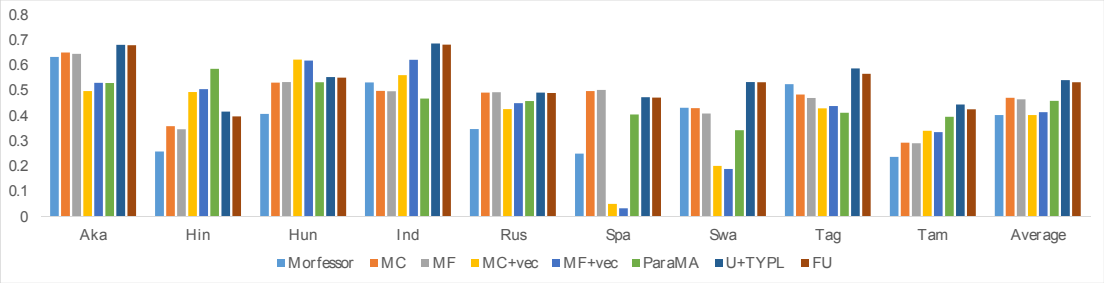

Figure 1: Comparison of different systems in F1 scores on the nine languages and their average. FU and U+T are our systems. FU is fully unsupervised, while U+T is unsupervised except given six flags for language typology.

	Morf	MC	MF	MC+v	MF+v	PMA	U+T	FU
Aka	0.633	0.650	0.646	0.498	0.530	0.530	**0.680**	**0.679**
Hin	0.258	0.359	0.346	0.494	0.505	**0.586**	0.432	0.398
Hun	0.407	0.532	0.533	**0.622**	0.619	0.532	0.554	0.551
Ind	0.532	0.499	0.497	0.561	0.622	0.469	**0.686**	**0.682**
Rus	0.347	0.492	**0.493**	0.427	0.450	0.458	0.493	0.490
Spa	0.250	0.498	**0.502**	0.051	0.034	0.405	0.473	0.472
Swa	0.432	0.430	0.409	0.202	0.189	0.343	**0.512**	**0.533**
Tag	0.525	0.484	0.470	0.430	0.439	0.411	**0.587**	**0.566**
Tam	0.237	0.293	0.291	0.341	0.336	0.396	**0.446**	**0.426**
Avg	0.402	0.471	0.465	0.403	0.414	0.459	**0.541**	**0.533**

Table 3: Experimental results in F1 measures on the nine languages including our unsupervised (FU) and oracle (U+T) system. The best score for each language is highlighted, considering each of our systems separately against previous work.

	Morf	MC	MF	MC+v	MF+v	PMA	U+T	FU
P	**0.618**	0.387	0.391	0.504	0.523	0.514	0.525	0.495
R	0.317	**0.647**	0.623	0.370	0.383	0.428	0.576	0.593
F1	0.402	0.471	0.465	0.403	0.414	0.459	**0.541**	0.533

Table 4: Average performance of the systems in precision, recall and F1 measures. Best result in bold, considering all systems together.

given the very low-resource setting, the vector configuration harms the performance of both MC and MF in languages such as Akan, Spanish, Swahili and Tagalog. Even though Russian has a larger corpus, the vectors still harm performance, which we believe is due to its complicated morphology that demands many examples to train reliable vectors.

While having separate patterns for each morphology type seems to improve numbers, oracle information improves results only slightly, mostly on Hindi, Tagalog, and Tamil. Interestingly, the performance on Swahili has been noticeably decreased. Based on detailed observation, this is due to our infixation search providing an unexpected benefit for Swahili, a language with no linguistic infixation, but the *final vowel* pattern, by allowing us to capture string-internal linguistic suffixes as

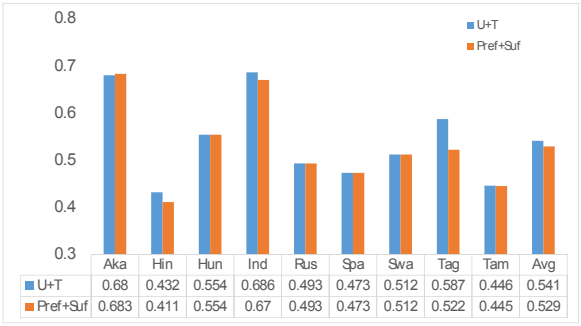

Figure 2: The performance of our model with oracle typological features (U+T) and with only prefixation and suffixation (Pref+Suf).

in the passive suffix *-w-* extracted from the verb *kunyang'anywa* here as *bx_<w>_ex*. In all, the performance of our model in either mode is better than the other systems we tested.

To test the contribution of morphological patterns other than prefixation and suffixation, we perform an ablation study, running the system with only prefixation and suffixation enabled. The results are shown in Figure 2. First, most performance for most languages is due to prefixation and suffixation since these are predominant for most languages. However, performance decreases measurably for Tagalog, Indonesian and Hindi due to the presence of more complex morphological patterns. This shows that modeling morphological features other than prefixation and suffixation has important benefits on languages with complicated morphology.

6.3 Discussion

Our system, in both its configurations, achieves the highest average performance among those tested. It has other advantages as well. Firstly, although our model is evaluated in terms of morpheme boundaries, it produces much richer structures than that.

It determines how a complex word is derived from another one through a particular morphological process such as prefixation, suffixation, infixation or full or partial reduplication. In comparison, other systems including Morpho-Chain, Morph-Forest, and ParaMA only deal with prefixes and suffixes. Our experiments as shown in Figure 2 indicate that modeling morphological patterns/processes other than prefixation and suffixation are useful.

Systems that directly find morpheme boundaries such as Morfessor are not aware of the particular morphological processes that a word's derivation goes through. So for infixed words, for example, even if the morpheme boundaries are correctly identified by such systems, they will incorrectly characterize the word as containing three morphemes rather than two. Such analyses are incorrect even though they are not penalized under a boundary-based evaluation metric.

By modeling different types of morphological structures, our system can be used to study the productivity of each morphological process and thus can be used for a quantitative analysis for theoretical morphological studies in linguistics. Figure 3 shows the number of instances of each type of morphological process generated by our fully unsupervised model. Suffixation and prefixation are the most common processes. Most of our test languages exhibit more suffixation than prefixation, but Swahili has more prefixation than suffixation, as expected for a Bantu language.

Figure 3 also shows that reduplication is rarer than other affixation. However, our model does discover full and left-partial reduplication successfully in languages that exhibit it. For example, about 1% of Akan words and fewer than 1% of Indonesian, Swahili and Tagalog words were analyzed with full or partial reduplication.

Infixation is challenging to correctly identify because infixes can appear in almost any position inside a word, and therefore generate a large search space. Our unsupervised system uses infixation to represent both true morphological infixation as in Tagalog as well as word-internal agglutinative suffixation as in Swahili, Hindi, and Tamil. This hurts the performance for Hindi and Tamil, but provides a benefit for Swahili as discussed above.

Finally, our system is fast, typically completing in several minutes, similar to ParaMA. Other systems including Morfessor, MC and MF typically require several hours, or even days on longer word lists such as for Hungarian and Russian.

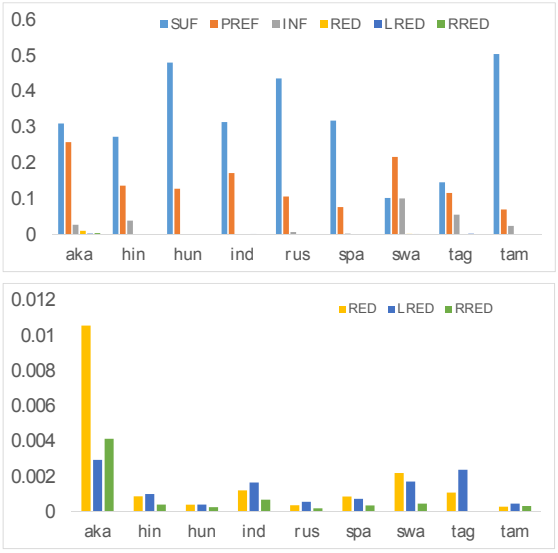

Figure 3: Normalized distribution of morphological patterns discovered by our unsupervised model for each language (top) and zoomed in on less frequent patterns (bottom). SUF: suffix, PREF: prefix, INF: infix, RED: full reduplication, LRED: left partial reduplication, RRED: right partial reduplication.

7 Conclusion and Future Work

In this paper, we develop a model for morphological analysis that exploits typological features to achieve the best performance on a wide range of languages. The tool is publicly available here: https://github.com/xuhongzhi/ParaMA2. This unsupervised model can be quickly and easily extended to novel languages without data annotation or expert input. Combined with the ability to process infixation and reduplication, our system improves access for geographically diverse low-resource languages. Although the evaluation is based on segmentation points, our model outputs much richer structure. It can also tell us the productivity of each morphological process and thus can obtain much deeper knowledge in terms of morphological structures of languages.

Our next step will be to attempt to automate the determination of language typology, yielding somewhat better performance with a system requiring no human intervention per language at all. Future work will aim to extend the current model to capture particularly challenging morphological patterns such as templatic non-concatenative morphology and polysynthetic composition.

Acknowledgements

We thank the rest of the University of Pennsylvania's LORELEI research team for the helpful discussions. We also thank the anonymous reviewers for their valuable and constructive comments for improving our paper. This research was funded by the DARPA LORELEI program under Agreement No. HR0011-15-2-0023.

References

Malin Ahlberg, Mans Hulden, and Markus Forsberg. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578.

Erwin Chan. 2006. Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, pages 69–78. Association for Computational Linguistics.

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

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

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

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

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

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(3):1–34.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 616–627. Association for Computational Linguistics.

Matthew S. Dryer. 2013. Prefixing vs. suffixing in inflectional morphology. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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

John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353–371.

Sharon Goldwater and Mark Johnson. 2004. Priors in bayesian learning of phonological rules. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 35–42. Association for Computational Linguistics.

Constantine Lignos. 2010. Learning from unseen data. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 35–38.

Jiaming Luo, Karthik Narasimhan, and Regina Barzilay. 2017. Unsupervised learning of morphological forests. *Transactions of the Association for Computational Linguistics*, 5:353–364.

Justin Mott, Ann Bies, Stephanie Strassel, Jordan Kodner, Caitlin Richter, Hongzhi Xu, and Mitchell Marcus. 2020. Morphological segmentation for low resource languages. In *International Conference on Language Resource and Evaluation (LREC)*.

Victor Mugari. 2013. Object marking restrictions on shona causative and applicative constructions. *Southern African Linguistics and Applied Language Studies*, 31(2):151–160.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Cornelia Parkes, Alexander M. Malek, and Mitchell P. Marcus. 1998. Towards unsupervised extraction of verb paradigms from large corpora. In *Proceedings of the Sixth Workshop on Very Large Corpora (COLING-ACL)*.

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

Carl Rubino. 2001. Pangasinan. In Jane Garry and Carl Rubino, editors, *Encyclopedia of the World's Languages: Past and Present*, pages 539–542. H.W. Wilson Press, New York / Dublin.

Carl Rubino. 2013. Reduplication. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics*, pages 737–745.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637.

Richard William Sproat. 1992. *Morphology and computation*. MIT press.

Gregory T Stump. 2001. *Inflectional morphology: A theory of paradigm structure*, volume 93. Cambridge University Press.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54.