Unsupervised Morphology Learning with Statistical Paradigms

Hongzhi Xu[1], **Mitch Marcus**[1], **Charles Yang**[2], **Lyle Ungar**[1]
[1] Computer and Information Science, University of Pennsylvania
[2] Linguistics Department, University of Pennsylvania
Philadelphia, PA 19104, USA
[1] {xh, mitch, ungar}@cis.upenn.edu
[2] charles.yang@ling.upenn.edu

Abstract

This paper describes an unsupervised model for morphological segmentation that exploits the notion of *paradigms*, which are sets of morphological categories (e.g., suffixes) that can be applied to a homogeneous set of words (e.g., nouns or verbs). Our algorithm identifies statistically reliable paradigms from the morphological segmentation result of a probabilistic model, and chooses reliable suffixes from them. The new suffixes can be fed back iteratively to improve the accuracy of the probabilistic model. Finally, the unreliable paradigms are subjected to pruning to eliminate unreliable morphological relations between words. The paradigm-based algorithm significantly improves segmentation accuracy. Our method achieves state-of-the-art results on experiments using the Morpho-Challenge data, including English, Turkish, and Finnish. [1]

1 Introduction

Morphological learning aims to automatically uncover constitutive units of words. It is an especially important task for many NLP applications such as language generation, information retrieval etc. (Sproat, 1992). Morphology analyzing is non-trivial especially for morphologically rich languages such as Turkish where the word formation process is extremely productive and can create in principle tens of billions of word forms. The identification of morphological relations between words provides a basis for uncovering their syntactic and semantic relations, which in turn can be exploited by downstream NLP applications.

Most unsupervised models of morphological segmentation (Virpioja et al., 2013; Goldwater and Johnson, 2004; Creutz and Lagus, 2005; Creutz and Lagus, 2007; Lignos, 2010; Poon et al., 2009; Snyder and Barzilay, 2008) treat words as concatenation of morphemes. In some models, the dependencies between morphemes (e.g., the English suffix *-es* often follows a verbal stem with *y* changed to *i*, as in *carries*) are recognized (Narasimhan et al., 2015), making use of transformations akin to rewrite rules (Goldwater and Johnson, 2004; Lignos et al., 2010). In all these approaches, the dependency between morphemes is generally local, and the overall distribution of the underlying paradigms implied by the segmentation result is not explored.

In this paper, we propose to exploit the notion of the *paradigm*, a global property of morphological systems, for the task of unsupervised morphological segmentation (Parkes et al., 1998; Goldsmith, 2001; Chan, 2006). The idea of using paradigm to describe the morphological structure of a language can be traced back to a long time ago, and has been widely adopted in modern linguistic studies, starting from Ferdinand de Saussure. A paradigm refers to a set of morphological categories such as suffixes that can be can be applied to a homogeneous class of words. For instance, the paradigm (*NULL, -er, -est, -ly*) is defined over English adjectives (e.g., *high, higher, highest, highly*), the paradigm (*NULL, -ing, -ed, -s, -er*) is defined over English verbs (e.g, *walk, walking, walked, walks, walker*), etc. In essence, a paradigm establishes an equivalence class for word formation such that a word realized in one of the categories in a paradigm can be expected to appear in all the categories in the paradigm.

[1] Code is available here: https://github.com/xuhongzhi/ParaMA

Proceedings of the 27th International Conference on Computational Linguistics, pages 44–54
Santa Fe, New Mexico, USA, August 20-26, 2018.

The advantages of using paradigms in morphological learning are manifold. On the one hand, paradigms provide a principled strategy for tackling the data sparsity problem. Not all morphologically possible forms of a word will be attested (Chan, 2006) and in a morphologically rich language such as Turkish, only a small fraction will be attested even in very large corpora. Paradigms can extend the attested morphological forms from few but high frequency words to low frequency words, likely the majority, for which there is little data. On the other hand, high quality paradigms may prove effective at detecting spurious morphological relations between words that have plagued many previous models. For instance, it is not uncommon for unsupervised morphological segmentation models to produce segmentations such as *with-in*, *with-out*, and *with-er*, where *with* is an attested word and *-in*, *-out*, and especially *-er*, are highly plausible suffixes (or more generally, morphemes). From the perspective of the paradigm, a global property defined over all words that take the suffix set (*-in, -er, -out*), it is clear that such a paradigm is very poorly supported—in fact by only one stem, namely, *with*, rather than a substantial set. This suffix set, then, is very unlikely to be a true paradigm and will be discarded, thereby eliminating segmentation errors such as *with-er*.

In this paper, we show that high quality morphological paradigms can be automatically constructed, resulting in considerable improvement in unsupervised morphological segmentation accuracy. Section 2 provides a review of previous and related work. Section 3 describes the general framework of our approach. Section 4 describes how to use linguistically-motivated language-independent heuristics to generate candidate segmentations with transformation rules for each word. Section 5 describes a probabilistic model of morphological learning that provides an initial segmentation including the identification of potential suffixes. Section 6 lays out the details of constructing morphological paradigms and a pruning process that eliminates spurious morphological relations. Section 7 reports the results of our experiments on Morpho-Challenge data including English, Turkish, and Finnish in comparison with previous models. Section 8 concludes with a discussion of future research.

2 Related Work

The Morpho-Challenge, held from 2005 to 2010, led to many successful morphology learning models. The Morfessor baseline system (Creutz and Lagus, 2002; Virpioja et al., 2013) provides a framework that maximizes the likelihood of the observation under the MDL principle. Creutz and Lagus (2005; 2007) extend the model with the maximum a posteriori (MAP) on both observed data and the model. Semi-supervised models have shown to be effective on morphological segmentation (Kohonen et al., 2010; Spiegler et al., 2010). In this paper, we focus on unsupervised learning of language morphologies, based on the consideration that constructing annotating data is expensive, especially for low-resource languages.

Narasimhan et al. (2015) adopt a log-linear model with semantic similarity measures obtained from word embedding to identify morphologically related word pairs (Schone and Jurafsky, 2001) and achieve impressive segmentation results on the Morpho-Challenge data. Such semantically based model, however, requires a large corpus to train reliable word embeddings, which renders the method unsuitable for low-resource languages.

The idea of paradigms has been explored in previous studies (Parkes et al., 1998; Goldsmith, 2001; Dreyer and Eisner, 2011; Ahlberg et al., 2014). Parkes et al. (1998) propose a model that learns neat inflectional paradigms only for English verbs from a corpus. Goldsmith (2001; 2006) uses heuristic rules with the MDL principle to greedily search morphological patterns (signatures). But the performance of rule-based search methods is crucially determined by the heuristic rules, and transformation rules are difficult to incorporate. Dreyer and Eisner (2011) proposed a log-linear model to identify paradigms. However, their method requires a number of seed paradigms for training. In morphologically rich languages such as Turkish, where one 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 somehow discovers inflection tables as an intermediate step.

In the paper, instead of constructing paradigms as a goal, we select statistically reliable paradigms from

the initial segmentation generated from a simple probabilistic model, and then use the reliable paradigms for pruning the unreliable ones, which we refer to paradigm pruning. The advantage of the proposed model is that it mathematically maximizes the likelihood of the observed data through the probabilistic model as well as maintains the global morphological structure in terms of paradigms. As will be demonstrated later, our method produces state-of-the-art results for morphological segmentation. It also provides a promising approach to unsupervised morphological learning for low-resource languages for which there is no sufficient quantity of data to enable embedding methods.

3 Our Method

We now formally describe our model. We write $w = (r, s, t)$ for a word w that consists of a root r, a suffix s, and a transformation rule t which captures stem changes in morphological processes. A morphologically simple word is treated as taking an empty suffix *NULL* without transformation rules. For example, the word *realizing* can be analyzed as deleting the last letter *e* from the word *realize* and adding suffix *-ing*, i.e. (*realize, -ing, DEL-e*). To deal with words with multiple suffixes is trivial. If $w = (r, s, t)$ and $r = (r', s', t')$, then $w = ((r', s', t'), s, t)$. Here, we call r the *immediate root* of w. If the word has itself as immediate root, i.e. taking a *NULL* suffix, it is called *atomic*. If r' is atomic, it is called the *final* root of w, otherwise, it is called an *intermediate* root of w. For example, the word *realizing* can be represented as ((*real, -ize, NULL*), *-ing*, *DEL-e*), where (*real, -ize, NULL*) represents the word *realize*. So, the final root of *realizing* is *real*, and the word *realize* is an intermediate root. Finally, the task of morphological segmentation for a word is to recursively find immediate root until its final root is found.

3.1 Modeling Transformation Rules

We model three stem changes, called transformation rules, namely *deletion*, *substitution*, and *duplication*, similarly to (Narasimhan et al., 2015). These three transformation rules were mainly designed to capture stem changes that are involved in suffixation. All transformation rules are represented with the specific characters involved in changes. The definitions of the three transformation rules are as follows.

1. **Deletion** (DEL) of the end letter of the root. For example, the word *using* can be analyzed as (*use, -ing, DEL-e*).

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

3. **Duplication** (DUP) of the end letter of the root. For example, the word *stopped* can be analyzed as (*stop, -ed, DUP+p*).

We note, however, that certain morphological phenomena do not readily yield to the transformation-based approach here. Infixation and templatic morphology are obvious examples. Even agglutinative systems, which at first glance appear suitable for transformation rules that operate at word edges, may still prove problematic when more global morphological processes are at play. For instance, the Turkish suffixes *-lar* and *-ler* will fall under two distinct transformational rules but are in fact one morpheme that is realized differently due to vowel harmony. This problem does not pose insurmountable problems for the purpose of morphological segmentation since both *-lar* and *-ler* are relatively frequent and can be identified as genuine (and distinct) suffixes, but clearly a more robust representation of morphological processes will be necessary to account for the full range of languages. We leave this problem for future research.

3.2 Morphological Segmentation Framework

Our method is schematically described in Algorithm 1. It has several major components. The GETPRIOR function sets the prior of the model parameters by assigning each candidate segmentation (r, s, t) of a word w equal probability. The function GENSEG generates candidate segmentations, (r, s, t), for each word w. A probabilistic model is then used to compute the probability of each candidate, i.e. $P(r, s, t)$

based on the parameters estimated by the function ESTIMATE. Then the segmentation with the maximum probability is chosen. The final segmentation (e.g. words with multiple suffixes) can be constructed recursively as described in the beginning of this section.

After that, the function PARADIGMS reorganizes the segmented words into paradigms. The function RELIABLE then selects a set of statistically reliable paradigms. The function ESTIMATE estimates the model parameters based on the segmentation result derived from reliable paradigms. Then the new parameters are used by the probabilistic model to get better segmentation result. The procedure iterates for several times. Here, we let the algorithm iterate twice as we find it sufficient to produce high quality segmentations. The function PRUNE prunes the unreliable paradigms. The final result is generated based on the reliable paradigms and the pruned ones with function SEGMENTATION. The following sections describe each component in details.

Algorithm 1 The main procedure

```
1: procedure MAIN(WordList D)
2:     {P(r, s, t)} ← GETPRIOR(D)
3:     while iter < maxIter do
4:         morph ← {}
5:         for all w in D do
6:             segs ← GENSEG(w)
7:             seg ← arg max_{(r,s,t)∈segs} P(r, s, t)
8:             morph ← morph + (w, seg)
9:         pdgs ← PARADIGMS(morph)
10:        pdgs_reliable, pdgs_unreliable ← RELIABLE(pdgs)
11:        {P(r, s, t)} ← ESTIMATE(pdgs_reliable)
12:    pdgs_pruned ← PRUNE(pdgs_unreliable)
13:    return SEGMENTATION(pdgs_pruned + pdgs_reliable)
```

4 Generating Candidate Segmentations

4.1 Selecting Candidate Suffixes

To obtain a working set of suffixes, we first adopt a simple method from previous studies: given a word pair (w_1, w_2), if $w_2 = w_1 + s$, then s is a candidate suffix (Keshava and Pitler, 2006; Dasgupta and Ng, 2007). By comparing all possible word pairs, we can generate a set of candidate suffixes with their counted frequencies. The more frequent a candidate is, the more likely it is to be a real suffix. In our system, we only keep candidate suffixes that are at most six character long and appear at least three times in the word list. If applied naively, this method produces many short, spurious suffixes that are frequently occurring substrings in words, e.g. (*for, fore*), (*are, area*), (*not, note*), (*she, shed*) etc. The problem can be overcome by imposing a minimum length on words that are subject to candidate suffix generation. In practice, we find that a minimum word length of four characters works well, which partially reflects the prosodic constraints on minimal words from the linguistic literature (McCarthy and Prince, 1999).

In addition, if a candidate suffix can be taken by words of various lengths, it is more likely to be a real one; if a candidate suffix can only be taken by short words, it is likely to be a false one. To further utilize this information, we use the following equation to calculate the confidence value (conf) of a candidate suffix.

$$conf(s) = \log(1 + |W_s|) \times \frac{1}{|W_s|} \times \sum_{w \in W_s} len(w) \quad (1)$$

where W_s is the set of words that can take the candidate suffix s and form a new word, and $len(w)$ is the length of word w. Finally, we can select the top N candidates.

4.2 Generating Candidate Segmentations with Transformation Rules

The procedure for generating a candidate segmentation of target word w begins by stripping a possible suffix s from the word. If the remaining part r is a valid word, then (r, s, *NULL*) is a possible candidate. If r is not a word, but there exists a word r'=r+c and $c \neq s$, then (r', s, *DEL-c*) is a candidate. If r is in

the form r'+c, and r' is a valid word also ending with c, then (r', s, *DUP-c*) is a candidate. If r is in the form r'+c, r' is not a word, but there exist another word w'=r'+c' and $c' \neq c$, then (w', s, *SUB-c+c'*) is a candidate. If no possible suffix s could be found, then add (w, *NULL*, *NULL*) as a candidate.

We apply the transformation rule types in the following ordering: (*NULL, duplication*, *deletion*, *substitution*). A candidate with a transformation rule is generated only if no candidate could be found with a previous type. The ordering reflects the linguistic approaches to morphology where suffixation applies to the stem changes, and the changes take place at morpheme boundaries before affecting the rest of the words on either side (Halle and Marantz, 1993). The linguistic reality of these processes, once more widespread in English, is now only latently reflected in modern English orthography but can be transparently observed in other languages.

5 A Probabilistic Model for Morphological Segmentation

We evaluate the conditional probability of a segmentation (r, s, t) given a word w. Since each triple (r, s, t) is uniquely associated with a single word w, denoted as $(r, s, t) = w$, for all (r, s, t), $P(r, s, t|w) = 0$ if $(r, s, t) \neq w$. Otherwise, we use the following formula to calculate this probability.

$$P(r, s, t|w) = \frac{P(r, s, t)}{\sum_{(r', s', t')=w} P(r', s', t')} \tag{2}$$

To compute $P(r, s, t)$, we assume that r is independent of s, and that t depends on both r and s. Taking into account that the transformation rules are not word form specific, but rather follow some constraints based on the phonological structures of the word and the suffix. We assume that the transformation rule t is dependent on feature extracted from r and s, denoted by $f(r, s)$. Thus, $P(r, s, t)$ can be decomposed as follows.

$$P(r, s, t) = P(r) \times P(s) \times P(t|f(r, s)) \tag{3}$$

In our implementation, we assume that t depends on the last character of the root r ($end(r)$) and the initial character of the suffix s ($init(s)$), i.e. $f(r, s) = end(r)$-$init(s)$. For example, the probability of the segmentation (*carry, -es, SUB-y-i*) for the word *carries* can be calculated by multiplying P(*carry*), P(*-es*), and P(*SUB-y+i|f*), where *f* is *y-e*. Again, this is an approximation of the morpho-phonological properties of language but one which nevertheless proves effective for morphological segmentation.

Finally, the segmentation of a word w can be predicted by choosing the segmentation (r, s, t) that maximizes $P(r, s, t)$ as follows.

$$seg(w) = \underset{(r,s,t)=w}{\operatorname{argmax}} P(r, s, t) \tag{4}$$

5.1 Parameter Estimation

To estimate the parameters $P(r)$, $P(s)$, $P(t|f)$, we initially assume that each candidate segmentation of a word has equal probability because the unsupervised model has no access to gold data. Each (r, s, t) of all possible segmentations seg of a word w then obtains $1/|seg|$ weight. After that, the probabilities $P(r)$, $P(s)$, and $P(t|f)$ can be easily computed based on the frequencies of r, s, f, and (t, f). This first estimation of the parameters $P(r)$, $P(s)$, $P(t|f)$ is the prior returned by the function GETPRIOR in Algorithm 1.

Consequently, the probability of a segmentation (r, s, t) of a word w can be computed using Formula 3, and select the segmentation with the maximum probability. Here, EM can also be used for estimating parameters, but we found that this simple method works very well. After the first round, the parameters can be re-estimated by only using the predicted segmentation of each word, with the function ESTIMATE in Algorithm 1.

As discussed above, the reliability of a candidate suffix is also related to the length of words that can take the suffix. So, another way of estimating $P(s)$ is to use the confidence value calculated with Equation 1. We will show that the method gives better results.

English		Turkish		Finnish	
Suffix Set	**Sup**	**Suffix Set**	**Sup**	**Suffix Set**	**Sup**
(-ed, -ing, -s)	772	(-ki, -n)	1560	(-ssa, -ta)	2465
(-ed, -ing)	331	(-nı, na)	207	(-en, -ta)	1132
(-ed, -er, -ing, -s)	219	(-nı, -na, -nda)	201	(-la, -le, -ta)	808
(-ly, -ness)	208	(-ne, -ni)	199	(-n, -ssa)	693
(-ed, -ing, -ion, -s)	154	(-ı, -a)	165	(-ssä, -tä)	677
(-ic, -s)	125	(-de, -e, -i, -in)	126	(-sen, -set, -sia, -ta)	462
(-ly, -s)	109	(-nde, -ne, -ni, -nin)	82	(-en, -ssa, -ta)	328
(-ed, -ing, -ment, -s)	63	(-dir, -ki, -n)	81	(-sen, -set, -siä, -tä)	177
(-ism, -s)	52	(-ğı, -tir)	81	(-a, -ksi, -la, -le, -ssa, -ta)	160
(-ed, -es, -ing)	52	(-de, -i)	79	(-aan, -ni)	156

Table 1: Examples of paradigm suffix sets and their supports of English, Turkish, and Finnish.

6 Paradigm Construction and Pruning

In this section, we discuss how to perform a post-pruning with a paradigm-based algorithm to exclude noisy segmentations. We define a paradigm formally as follows.

- **Paradigm** $p = S \times R$ is a Cartesian product of a set of suffix $S = \{s_i\}$ and a set of roots $R = \{r_i\}$, such that for any suffix $s \in S$ and root $r \in R$, r can take s to form a valid word w by applying a transformation rule t [2].

With this definition, the larger the cardinality $|R|$ is, the more reliable a paradigm p is. On the other hand, it is not always true that the larger $|S|$ is, the more reliable a paradigm is. An extreme case occurs when there is only one root in the paradigm, i.e. $|R| = 1$. For example, the root *the* forms a paradigm with 42 possible suffixes and all of them are false. We therefore define the support of a paradigm as follows.

- **Support** (SUP) of the a paradigm $p = S \times R$ is $|R|$, the cardinality of the root set R.

6.1 Constructing Paradigms

After we get the segmentation (r, s, t) for each word w, the paradigms can be easily obtained by grouping together the words that share the same immediate root r, regardless of the transformation rules that are involved. For example, the words *reporting*, *reported*, and *reports* are segmented as (*report, -ed, NULL*), (*report, -ing, NULL*), and (*report, s, NULL*) respectively, and the words *baked*, *baking*, *bakes* are segmented as (*bake, -ed, DEL-e*), (*bake, -ing, DEL-e*), and (*bake, -s, NULL*). Then we can construct a paradigm based on these two words as {*-ed, -ing, -s*} × {*report, bake*}.

6.2 Paradigm Pruning

We crucially assume that even though the segmentation result given by the initial model is not highly accurate, the distribution of paradigms constructed will provide clear evidence of whether they are reliable. Table 1 shows some examples of paradigms with more than one suffix, and these are indeed consistent with the morphological structures of the languages.

The paradigms with more than one suffix and with support value larger than 1 are selected as the reliable ones, the same strategy used by Goldsmith (2001) for filtering when constructing candidate paradigms. The method has two consequences. First, we exclude a large proportion of suffixes that only appear in unreliable paradigms so that the frequency of suffixes be estimated based on the reliable paradigms. Second, we can use the reliable paradigms as references for pruning the unreliable ones.

The basic idea of pruning unreliable paradigms with reliable ones is to take the intersection of the suffix set of a paradigm to be pruned and that of any of the reliable ones and to choose the one that achieves the best score. The score of a set of suffixes is calculated according to the following equation.

[2]Transformation rules (or stem changes) are not considered a part of the paradigms, because they are usually not directly driven by morphological processes but rather some phonological rules or others

	English	Turkish	Finnish
Training (MC:10)	878,036	617,298	2,928,030
Test (MC:05-10)	2,218	2,534	2,495

Table 2: Data Set. MC:10 is the Morpho-Challenge 2010 and MC:05-10 is the combined data of Morpho-Challenge 2005-2010.

$$score(S) = \sum_{s \in S} conf(s) \tag{5}$$

For instance, suppose we have an unreliable paradigm {*-ed, -ing, -s, -se*} × {*appear*} [3], which only has support value 1, and there are two reliable paradigms with suffix sets (*-ed, -ing, -s*) and (*-ed, -ing*) respectively. Then, the pruning algorithm calculates the intersection of the unreliable one and each of the reliable ones, resulting in (*-ed, -ing, -s*) and (*-ed, -ing*). According to Equation 5, the former one will be kept since it has a higher score. Thus, the original paradigm is pruned to {*-ed, -ing, -s*} × {*appear*}, and consequently, the false morphological relation between *appease* and *appear* (by *-se*) is filtered out.

7 Experiments

7.1 Experiment Setting

7.1.1 Data

We ran experiments on a combined version of the Morpho-Challenge 2005-2010 data sets including English, Turkish, and Finnish, the same setup as Narasimhan et al. (2015). In our experiments, testing words are included in the training set since the method is unsupervised. A statistical description of the data is shown in Table 2.

As in previous work on unsupervised morphological learning, we use a frequency-based filtering method to reduce noise in the data. This is necessary because the word list provided by Morpho-Challenge 2010 is generated by lower-casing all running tokens in the corpora including abbreviations and proper nouns. Many of these are three characters words. We use the following conditions to select reliable roots: 1) $freq \geq 2000$ if $len(word) \leq 3$; 2) $freq \geq 200$ if $len(word) \leq 4$; 3) $freq \geq 20$ if $len(word) \leq 5$; 4) $freq \geq 3$ else. The motivation is that short words are expected to be more frequent; rare short words are likely to be abbreviations or noise.

7.1.2 Compounding

We also add a compound inference module, which simply splits a word w before generating candidate segmentations if it is composed of w_1 and w_2. If there is more than one possible segmentation, then we choose the one with maximum length of w_1. The candidate segmentations are then generated for w_1 and w_2 separately. The final segmentation result is obtained by combining the segmentation results of w_1 and w_2.

7.1.3 Evaluation

Following (Narasimhan et al., 2015), we measure the performance of our model with segmentation points, i.e. the boundaries between morphemes in words. The precision, recall and F1 values on the identification of segmentation points are reported.

7.2 Experiment Results

7.2.1 Ablation test

We first test five different variations of our model. The first one is the baseline system (Base) which only uses the probabilistic Model without transformation rules. The second one (+Trans) is the baseline plus transformation rules. The third one (+Comp) is the second one plus the compounding inference module. The fourth one (+Prune) is the third one plus the paradigm pruning algorithm. The fifth one (+Conf) is

[3]The word *appear* takes suffix *-se* to form *appease* through a transformation rule *DEL-r*.

	English			Turkish			Finnish		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Base	0.483	0.686	0.567	0.616	0.621	0.619	0.521	0.245	0.333
+Trans	0.531	0.807	0.641	0.589	0.728	0.651	0.393	0.338	0.363
+Comp	0.511	0.861	0.641	0.582	0.728	0.647	0.389	0.606	0.474
+Prune	0.814	0.783	**0.798**	0.651	0.514	0.574	0.688	0.436	0.534
+Conf	0.810	0.787	**0.798**	0.600	0.746	**0.665**	0.712	0.481	0.574

Table 3: Experimental result of our model.

	English			Turkish			Finnish		
Method	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Morf-Base	0.740	0.623	0.677	0.827	0.362	0.504	0.839	0.357	0.501
Morf-Cat	0.673	0.587	0.627	0.522	0.607	0.561	0.782	0.452	0.573
LogLinear-C	0.555	0.792	0.653	0.516	0.652	0.576	0.483	0.650	0.554
LogLinear-Full	0.807	0.722	0.762	0.743	0.520	0.612	0.428	0.496	0.460
Our model	0.810	0.787	**0.798**	0.600	0.746	**0.665**	0.824	0.452	**0.584**

Table 4: Comparison of our model with others. The numbers for Finnish are obtained by running the systems by ourselves. The other numbers are from (Narasimhan et al., 2015).

the same as the fourth one except that the estimation of $P(s)$ is based on the confidence value calculated through Equation 1. In order to achieve the best performance, if a feature harms the performance, it will be removed in the next round.

The results are shown in Table 3. Firstly, we can see that incorporating of transformation rules improves the performance for all the three languages with 7.6%, 3.2%, and 3.0% improvements of F1 measure respectively. After adding the compounding analysis module, the performance is significantly improved on Finnish, with 11.1% improvement of F1 measure. The paradigm pruning algorithm significantly improves the performance on English and Finnish, with 15.7% and 6.0% of F1 measure, and improves precision of the model on Turkish, although the overall performance decreases. Finally, by using confidence based estimation of $P(s)$, the performance is improved further on all the three languages, achieving the best result for English and Turkish. For Finnish, the best result is actually achieved without transformation rules, namely 0.824, 0.452, and 0.584 in Precision, Recall, and F1 respectively.

7.2.2 Comparison with other models

We compare our model with three systems including the Morfessor Baseline system (Morf-Base) (Virpioja et al., 2013), Morfessor CatMAP (Morf-Cat), and the Log-linear model with full features (LogLinear-Full) and the model without semantic similarity (LogLinear-C) in (Narasimhan et al., 2015). Besides English and Turkish as used in (Narasimhan et al., 2015), we also add Finnish for experiments. For training word embeddings which will be in LogLinear-Full model, we use a corpus created in the DARPA LORELEI [4] project, which contains about 101 million tokens.

The result is shown in Table 4. The numbers for English and Turkish are from (Narasimhan et al., 2015). We can see that our model achieves the best performance in all the three languages. The LogLinear-Full model is the second best model on English and Turkish. However, it is worth noting that that model is based on semantic similarity features which requires training word vectors on independent corpora. Our model, on the other hand, only uses a list of words. The word frequencies are only used to filter noise. Our model is significantly better than the (Narasimhan et al., 2015) model without semantic embedding (i.e. LogLinear-C), with 20.1% relative improvement of F1 on English and 12.8% relative improvement on Turkish. For Finnish, the Morfessor CatMAP model has similar result as ours. The semantic similarity information harms the LogLinear-Full model. We think that this is due to the data sparseness problem as Finnish is a highly synthetic language, which then requires a larger corpus for training effective word embeddings.

[4] https://www.darpa.mil/program/low-resource-languages-for-emergent-incidents

7.3 Error Analysis and Discussion

Morphology learning systems in general suffer from two major problems, namely over-segmentation and under-segmentation. Over-segmentation is usually caused by spurious roots, either intermediate or final, such as the over-segmented words caused by the short frequent words, e.g. *the, with*, etc. Under-segmentation is usually caused by unseen intermediate roots or unidentified real suffixes. In morphologically rich languages like Turkish and Finnish where a root can take multiple suffixes, this problem is more serious.

Firstly, we can see from Table 3 that the use of transformation rules significantly reduces the under-segmentation problem in all the three languages as indicated by the increased recall rates. For English, the transformation rules also reduce the over-segmentation problem. This is due to fact that the transformation rules can well capture the morphology of English and thus significantly increase the true positive segmentations. On the other hand, the transformation rules increase the over-segmentation problem for Turkish and Finnish as indicated by the decreased precisions. For Finnish, the best performance of the system is achieved without transformation rules. That is due to the fact that Finnish morphology involves a large number of vowel changes, e.g. lengthening and shortening at non-boundary positions, which cannot be captured by the current transformation rules. However, the transformation rules we use introduce a large number of false stem changes which then causes the over-segmentation problem increased. We will address this problem in our future research.

Secondly, the compounding module can further improve the recall rates and thus decrease the under-segmentation problem for English and Finnish. This also reflects the compounding nature of the languages and also the distribution of the test set.

Finally, the pruning algorithm significantly reduces over-segmentations for all three languages as indicated by the increased precision values. However, the under-segmentation problem is also increased. This is due to the identification of incomplete paradigms which is then caused by data sparseness problem. For Turkish, the problem is even more serious. Pruning with incomplete paradigms will falsely exclude real suffixes from an unreliable paradigm. This problem can be potentially addressed by merging proper paradigms to identify the maximal suffix sets (complete paradigms). This will be in our future research.

8 Conclusion and Future Work

In this paper, we propose an unsupervised model of morphology learning which outperforms the state-of-the-art systems, using only orthographic information from a word list. Our contribution also lies in providing a new method of using automatically learned paradigms to fine tune the morphological segmentation results produced by a simple probabilistic model. This method is effective in eliminating spurious segmentations and improving the segmentation accuracy. Finally, we also use the word length information to select good candidate suffixes and estimate the suffix probabilities, which can further improve the performance of the model. In addition, combining our model and semantics based systems can potentially yield better result since they use different kinds of information and complement each other.

We believe that our approach of using paradigms provides a foundation for dealing with other morphological types such as prefixes, infixes, reduplication etc. In detail, the notation of the suffix variable s can be generalized to f, a morphological function that takes a root as input and produces the derived form. Correspondingly, the definition of paradigms could be easily revised as a set of morphological functions that can take a set of words and generate their derived forms. We will work on extending our system to process other types of morphologies in our future research.

Acknowledgements

We thank the rest of the University of Pennsylvanias LORELEI research team for the helpful discussions. We also thank the anonymous reviewers who have given valuable and constructive comments as well as insightful suggestions for improving our system. 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.

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.

Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *HLT-NAACL*, pages 155–163.

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.

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.

Morris Halle and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Kenneth Hale and Samuel Jay Keyser, editors, *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*, pages 111–176. MIT Press, Cambridge, MA.

Samarth Keshava and Emily Pitler. 2006. A simpler, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 31–35.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86. Association for Computational Linguistics.

Constantine Lignos, Erwin Chan, Mitchell P. Marcus, and Charles Yang. 2010. A rule-based acquisition model adapted for morphological analysis. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 658–665. Springer.

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

John J McCarthy and Alan Prince. 1999. Faithfulness and identity in prosodic morphology. *The prosody-morphology interface*, 79:218–309.

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.

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.

Sebastian Spiegler, Bruno Golénia, and Peter A. Flach. 2010. Word decomposition with the promodes algorithm family bootstrapped on a small labelled dataset. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 49–52.

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

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