Appeared in the proceedings of CoNLL 2015 (Beijing, July). This version was prepared in June 2018 and is a slightly clarified version of the published version.

Labeled Morphological Segmentation with Semi-Markov Models

Ryan Cotterell[1,2] **Thomas Müller**[2]
Department of Computer Science[1]
Johns Hopkins University, USA
ryan.cotterell@jhu.edu

Alexander Fraser[2] **Hinrich Schütze**[2]
Center for Information and Language Processing[2]
University of Munich, Germany
muellets@cis.lmu.de

Abstract

We present labeled morphological segmentation, an alternative view of morphological processing that unifies several tasks. We introduce a new hierarchy of morphotactic tagsets and CHIPMUNK, a discriminative morphological segmentation system that, contrary to previous work, explicitly models morphotactics. We show improved performance on three tasks for all six languages: (i) morphological segmentation, (ii) stemming and (iii) morphological tag classification. For morphological segmentation our method shows absolute improvements of 2-6 points F_1 over the baseline.

1 Introduction

Morphological processing is often an overlooked problem since many well-studied languages (e.g., Chinese and English) are morphologically impoverished. But for languages with complex morphology (e.g., Finnish and Turkish) morphological processing is essential. A specific form of morphological processing, morphological segmentation, has shown its utility for machine translation (Dyer et al., 2008), sentiment analysis (Abdul-Mageed et al., 2014), bilingual word alignment (Eyigöz et al., 2013), speech processing (Creutz et al., 2007b) and keyword spotting (Narasimhan et al., 2014), inter alia. We advance the state-of-the-art in supervised morphological segmentation by describing a high-performance, data-driven tool for handling complex morphology, even in low-resource settings.

In this work, we make the distinction between *unlabeled morphological segmentation* (*UMS*) (often just called "morphological segmentation") and *labeled morphological segmentation* (LMS). The labels in our supervised discriminative model for LMS capture the distinctions between different types of morphemes and directly model the morphotactics. We further create a hierarchical universal tagset for labeling morphemes, with different levels appropriate for different tasks. Our hierarchical tagset was designed by creating a standard representation from heterogeneous resources for six languages. This allows us to use a *single unified framework* to obtain strong performance on three common morphological tasks that have typically been viewed as *separate problems* and addressed using *different methods*. We give an overview of the tasks addressed in this paper in Figure 1. The figure shows the expected output for the Turkish word *gençleşmelerin* 'of rejuvenatings'. In particular, it shows the full labeled morphological segmentation, from which three representations can be directly derived: the unlabeled morphological segmentation, the stem/root[1] and the morphological tag containing POS and inflectional features.

We model these tasks with CHIPMUNK, a semi-Markov conditional random field (semi-CRF) (Sarawagi and Cohen, 2004), a model that is well-suited for morphology. We provide an evaluation and analysis on six languages; CHIPMUNK yields strong results on all three tasks, including state-of-the-art accuracy on morphological segmentation.

Paper Outline. Section 2 presents our LMS framework and the morphotactic tagsets we use, i.e., the labels of the sequence prediction task CHIPMUNK solves. Section 3 introduces our semi-CRF model. Section 4 presents our novel features. Section 5 compares CHIPMUNK to previous work. Section 6 presents experiments on the three complementary tasks of segmentation (UMS), stemming, and morphological tag classification. Section 7

[1] Terminological notes: We use *root* to refer to a morpheme with concrete meaning, *stem* to refer to the concatenation of all roots and derivational affixes, *root detection* to refer to stripping both derivational *and* inflectional affixes, and *stemming* to refer to stripping only inflectional affixes.

gençleşmelerin					
UMS	genç	leş	me	ler	in
Gloss	young	-ate	-ion	-s	GENITIVE MARKER
LMS	genç ROOT:ADJECTIVAL	leş SUFFIX:DERIV:VERB	me SUFFIX:DERIV:NOUN	ler SUFFIX:INFL:NOUN:PLURAL	in SUFFIX:INFL:NOUN:GENITIVE
Root	genç	**Stem**	gençleşme	**Morphological Tag**	PLURAL:GENITIVE

Figure 1: Examples of the tasks addressed for the Turkish word *gençleşmelerin* 'of rejuvenatings': Traditional unlabeled segmentation (UMS), Labeled morphological segmentation (LMS), stemming / root detection and (inflectional) morphological tag classification. The morphotactic annotations produced by LMS allow us to solve these tasks using a single model.

briefly discusses finite-state morphology.

The datasets created, additional description of our tagsets and CHIPMUNK can be found at http://cistern.cis.lmu.de/chipmunk.

2 Labeled Segmentation and Tagset

We define the framework of *labeled morphological segmentation* (LMS), an enhancement of morphological segmentation that—in addition to identifying the boundaries of segments—*assigns a fine-grained morphotactic tag to each segment*. LMS leads to both better modeling of segmentation and subsumes several other tasks, e.g., stemming.

Most previous approaches to morphological segmentation are either unlabeled or use a small, coarse-grained set such as prefix, root, suffix. In contrast, our labels are fine-grained. This finer granularity has two advantages. (i) The labels are needed for many tasks, for instance in sentiment analysis detecting morphologically encoded negation, as in Turkish, is crucial. In other words, for many applications UMS is insufficient. (ii) The LMS framework allows us to learn a probabilistic model of morphotactics. Working with LMS results in higher UMS accuracy. So even in applications that only need segments and no labels, LMS is beneficial. Note that the concatenation of labels *across* segments yields a bundle of morphological attributes similar to those found in the CoNLL datasets often used to train morphological taggers (Buchholz and Marsi, 2006)—thus LMS helps to unify UMS and morphological tagging. We believe that LMS is a needed extension of current work in morphological segmentation. Our framework concisely allows the model to capture interdependencies among various morphemes and model relations between entire morpheme classes—a neglected aspect of the problem.

We first create a hierarchical tagset with increasing granularity, which we created by analyzing the heterogeneous resources for the six languages we work on. The optimal level of granularity is task and language dependent: the level is a trade-off between simplicity and expressivity. We illustrate our tagset with the decomposition of the German word *Enteisungen* 'defrostings' (Figure 2).

The level 0 tagset involves a single tag indicating a segment. It ignores morphotactics completely and is similar to previous work. The level 1 tagset crudely approximates morphotactics: it consists of the tags {PREFIX, ROOT, SUFFIX}. This scheme has been successfully used by unsupervised segmenters, e.g., MORFESSOR CAT-MAP (Creutz et al., 2007a). It allows the model to learn simple morphotactics, for instance that a prefix cannot be followed by a suffix. This makes a decomposition like *reed* → *re+ed* unlikely. We also add an additional UNKNOWN tag for morphemes that do not fit into this scheme. The level 2 tagset splits affixes into DERIVATIONAL and INFLECTIONAL, effectively increasing the maximal tagset size from 4 to 6. These tags can encode that many languages allow for transitions from derivational to inflectional endings, but rarely the opposite. This makes the incorrect decomposition of German *Offenheit* 'openness' into *Off*, inflectional *en* and derivational *heit* unlikely[2]. This tagset is also useful for building statistical stemmers. The level 3 tagset adds POS, i.e., whether a root is VERBAL, NOMINAL or ADJECTIVAL, and the POS of the word that an affix derives. The level 4 tagset includes the inflectional feature a suffix adds, e.g., CASE or NUMBER. This is helpful for certain agglutinative languages, in which, e.g., CASE must follow NUMBER. The level 5 tagset adds the actual value of the inflectional feature, e.g., PLURAL, and corresponds to the annotation in the datasets. In preliminary experiments we found that the level 5 tagset is too rich

[2]Like *en* in English *open*, *en* in German *Offen* is part of the root.

5	PREFIX:DERIV:VERB	ROOT:NOUN	SUFFIX:DERIV:NOUN	SUFFIX:INFL:NOUN:PLURAL
4	PREFIX:DERIV:VERB	ROOT:NOUN	SUFFIX:DERIV:NOUN	SUFFIX:INFL:NOUN:NUMBER
3	PREFIX:DERIV:VERB	ROOT:NOUN	SUFFIX:DERIV:NOUN	SUFFIX:INFL:NOUN
2	PREFIX:DERIV	ROOT	SUFFIX:DERIV	SUFFIX:INFL
1	PREFIX	ROOT	SUFFIX	SUFFIX
0	SEGMENT	SEGMENT	SEGMENT	SEGMENT
German	Ent	eis	ung	en
English	de	frost	ing	s

Figure 2: Example of the different morphotactic tagset granularities for German *Enteisungen* 'defrostings'.

level:	0	1	2	3	4
English	1	4	5	13	16
Finnish	1	4	6	14	17
German	1	4	6	13	17
Indonesian	1	4	4	8	8
Turkish	1	3	4	10	20
Zulu	1	4	6	14	17

Table 1: Morphotactic tagset size at each level of granularity.

and does not yield consistent improvements, we thus do not explore it. Table 1 shows tagset sizes for the six languages.[3]

3 Model

CHIPMUNK is a supervised model implemented using the well-understood semi-Markov conditional random field (semi-CRF) (Sarawagi and Cohen, 2004) that naturally fits the task of LMS. Semi-CRFs generalize linear-chain CRFs and model segmentation jointly with sequence labeling. Just as linear-chain CRFs are discriminative adaptations of *hidden Markov models* (Lafferty et al., 2001), semi-CRFs are an analogous adaptation of *hidden semi-Markov models* (Murphy, 2002). Semi-CRFs allow us to integrate new features that look at complete segments, this is not possible with CRFs, making semi-CRFs a natural choice for morphology.

A semi-CRF represents $\boldsymbol{w}$ (a word) as a sequence of segments $\boldsymbol{s} = \langle s_1, \ldots, s_n \rangle$, each of which is assigned a label ℓ_i. The concatenation of all segments equals $\boldsymbol{w}$. We seek a log-linear distribution $p_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\ell} \mid \boldsymbol{w})$ over all possible segmentations and label sequences for $\boldsymbol{w}$, where $\boldsymbol{\theta}$ is the parameter vector. Note that we recover the standard CRF if we restrict the segment length to 1. Formally, we define $p_{\boldsymbol{\theta}}$ as

$$p_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\ell} \mid \boldsymbol{w}) \stackrel{\text{def}}{=} \frac{1}{Z_{\boldsymbol{\theta}}(\boldsymbol{w})} \prod_{i=1}^{n} e^{\boldsymbol{\theta}^T \boldsymbol{f}(s_i, \ell_i, \ell_{i-1}, i)}, \quad (1)$$

[3] As converting segmentation datasets to tagsets is not always straightforward, we include tags that lack some features, e.g., some level 4 German tags lack POS because our German data does not specify it.

where $\boldsymbol{f}$ is the feature function and $Z_{\boldsymbol{\theta}}(\boldsymbol{w})$ is the partition function. To keep the notation uncluttered, we will write $\boldsymbol{f}$ without all its arguments in the future. We use a generalization of the forward-backward algorithm for efficient gradient computation (Sarawagi and Cohen, 2004). Inspection of the semi-Markov forward recursion,

$$\boldsymbol{\alpha}(t, l) = \sum_{i=1}^{t-1} \sum_{\ell'=1}^{L} e^{\boldsymbol{\theta}^T \boldsymbol{f}} \cdot \boldsymbol{\alpha}(t - i, \ell'), \quad (2)$$

shows that algorithm runs in $\mathcal{O}(n^2 \cdot L^2)$ time where n is the length of the word $\boldsymbol{w}$ and L is the number of labels (size of the tagset).

We employ the maximum-likelihood criterion to estimate the parameters with L-BFGS (Liu and Nocedal, 1989), a gradient-based optimization algorithm. As in all exponential family models, the gradient of the log-likelihood takes the form of the difference between the observed and expected features counts (Wainwright and Jordan, 2008) and can be computed efficiently with the semi-Markov extension of the forward-backward algorithm. We use L_2 regularization with a regularization coefficient tuned during cross-validation.

We note that semi-Markov models have the potential to obviate typical errors made by standard Markovian sequence models with an IOB labeling scheme over characters. For instance, consider the incorrect segmentation of the English verb *sees* into *se+es*. These are reasonable split positions as many English stems end in *se* (e.g., consider *abuse-s*). Semi-CRFs have a major advantage here as they can have *segmental* features that allow them to learn *se* is not a good morph.

4 Features

We introduce several novel features for LMS. We exploit existing resources, e.g., spell checkers and Wiktionary, to create straightforward and effective features and we incorporate ideas from related areas: named-entity recognition (NER) and morphological tagging.

	# Affixes	Random Examples
English	394	-ard -taxy -odon -en -otic -fold
Finnish	120	-tä -llä -ja -t -nen -hön -jä -ton
German	112	-nomie -lichenes -ell -en -yl -iv
Indonesian	5	-kau -an -nya -ku -mu
Turkish	263	-ten -suz -mek -den -t -ünüz
Zulu	72	i- u- za- tsh- mi- obu- olu-

Table 2: Sizes of the various affix gazetteers.

Affix Features and Gazetteers. In contrast to syntax and semantics, the morphology of a language is often simple to document and a list of the most common morphs can be found in any good grammar book. Wiktionary, for example, contains affix lists for all the six languages used in our experiments.[4] Providing a supervised learner with such a list is a great boon, just as gazetteer features aid NER (Smith and Osborne, 2006)—perhaps even more so since suffixes and prefixes are generally *closed-class*; hence these lists are likely to be comprehensive. These features are binary and fire if a given substring occurs in the gazetteer list. In this paper, we simply use suffix lists from English Wiktionary, except for Zulu, for which we use a prefix list, see Table 2.

We also include a feature that fires on the conjunction of tags and substrings observed in the training data. In the level 5 tagset this allows us to link all allomorphs of a given morpheme. In the lower level tagsets, this links related morphemes. Virpioja et al. (2010) explored this idea for unsupervised segmentation. Linking allomorphs together under a single tag helps combat sparsity in modeling the morphotactics.

Stem Features. A major problem in statistical segmentation is the reluctance to posit morphs not observed in training; this particularly affects roots, which are *open-class*. This makes it nearly impossible to correctly segment compounds that contain unseen roots, e.g., to correctly segment *homework* you need to know that *home* and *work* are independent English words. We solve this problem by incorporating spell-check features: binary features that fire if a segment is valid for a given spell checker. Spell-check features function effectively as a proxy for a "root detector". We use the open-source ASPELL dictionaries as they are freely available in 91 languages. Table 3 shows the coverage of these dictionaries.

[4] A good example of such a resource is `en.wiktionary.org/wiki/Category:Turkish_suffixes`.

English	119,839
Finnish	6,690,417
German	364,564
Indonesian	35,269
Turkish	80,261
Zulu	73,525

Table 3: Number of words covered by the respective ASPELL dictionary

Integrating the Features. Our model uses the features discussed in this section and additionally the simple n-gram context features of Ruokolainen et al. (2013). The n-gram features look at variable length substrings of the word on both the right and left side of each boundary. We create conjunctive features from the cross-product between the morphotactic tagset (Section 2) and the features.

5 Related Work

Van den Bosch and Daelemans (1999) and Marsi et al. (2005) present memory-based approaches to discriminative learning of morphological segmentation. This is the previous work most similar to our work. They address the problem of LMS. We distinguish our work from theirs in that we define a cross-lingual schema for defining a hierarchical tagset for LMS. Morever, we tackle the problem with a feature-rich log-linear model, allowing us to easily incorporate disparate sources of knowledge into a single framework, as we show in our extensive evaluation.

UMS has been mainly addressed by unsupervised algorithms. LINGUISTICA (Goldsmith, 2001) and MORFESSOR (Creutz and Lagus, 2002) are built around an idea of optimally encoding the data, in the sense of minimal description length (MDL). MORFESSOR CAT-MAP (Creutz et al., 2007a) formulates the model as sequence prediction based on HMMs over a morph dictionary and MAP estimation. The model also attempts to induce basic morphotactic categories (PREFIX, ROOT, SUFFIX). Kohonen et al. (2010a,b) and Grönroos et al. (2014) present variations of MORFESSOR for semi-supervised learning. Poon et al. (2009) introduces a Bayesian state-space model with corpus-wide priors. The model resembles a semi-CRF, but dynamic programming is no longer possible due to the priors. They employ the three-state tagset of Creutz and Lagus (2004) (row 1 in Figure 2) for Arabic and Hebrew UMS. Their gradient and objective computation is based on an enumeration of

	Un. Data	Train+Tune+Dev			Test
		Train	Tune	Dev	
English	878k	800	100	100	694
Finnish	2,928k	800	100	100	835
German	2,338k	800	100	100	751
Indonesian	88k	800	100	100	2500
Turkish	617k	800	100	100	763
Zulu	123k	800	100	100	9040

Table 4: Dataset sizes (number of types).

a heuristically chosen subset of the exponentially many segmentations. This limits its applicability to language with complex *concatenative* morphology, e.g., Turkish and Finnish.

Ruokolainen et al. (2013) present an averaged perceptron (Collins, 2002), a discriminative structured prediction method, for UMS. The model outperforms the semi-supervised model of Poon et al. (2009) on Arabic and Hebrew morpheme segmentation as well as the semi-supervised model of Kohonen et al. (2010a) on English, Finnish and Turkish.

Finally, Ruokolainen et al. (2014) get further consistent improvements by using features extracted from large corpora, based on the letter successor variety (LSV) model (Harris, 1995) and on unsupervised segmentation models such as Morfessor CatMAP (Creutz et al., 2007a). The idea behind LSV is that for example *talking* should be split into *talk* and *ing*, because *talk* can also be followed by different letters then *i* such as *e* (talked) and *s* (talks).

Chinese word segmentation (CWS) is related to UMS. Andrew (2006) successfully apply semi-CRFs to CWS. The problem of joint CWS and POS tagging (Ng and Low, 2004; Zhang and Clark, 2008) is related to LMS. To our knowledge, joint CWS and POS tagging has not been addressed by a simple single semi-CRF, possibly because POS tagsets typically used in Chinese treebanks are much bigger than our morphotactic tagsets and the morphological poverty of Chinese makes higher-order models necessary and the direct application of semi-CRFs infeasible.

6 Experiments

We experimented on six languages from diverse language families. The segmentation data for English, Finnish and Turkish was taken from MorphoChallenge 2010 (Kurimo et al., 2010).[5] Despite typically being used for UMS tasks, the MorphoChallenge datasets do contain morpheme level labels. The German data was extracted from the CELEX2 collection (Baayen et al., 1993). The Zulu data was taken from the Ukwabelana corpus (Spiegler et al., 2010). Finally, the Indonesian portion was created applying the rule-based analyzer MORPHIND (Larasati et al., 2011) to the Indonesian portion of an Indonesian-English bilingual corpus.[6]

We did not have access to the MorphoChallenge test set and thus used the original development set as our final evaluation set (Test). We developed CHIPMUNK using 10-fold cross-validation on the 1000 word training set and split every fold into training (Train), tuning (Tune) and development sets (Dev).[7] For German, Indonesian and Zulu we randomly selected 1000 word forms as training set and used the rest as evaluation set. For our final evaluation we trained CHIPMUNK on the concatenation of Train, Tune and Dev (the original 1000 word training set), using the optimal parameters from the cross-evaluation and tested on Test.

One of our baselines also uses unlabeled training data. MorphoChallenge provides word lists for English, Finnish, German and Turkish. We use the unannotated part of Ukwabelana for Zulu; and for Indonesian, data from Wikipedia and the corpus of Krisnawati and Schulz (2013).

Table 4 shows the important statistics of our datasets.

In all evaluations, we use variants of the standard MorphoChallenge evaluation approach. Importantly, for word types with multiple correct segmentations, this involves finding the maximum score by comparing our hypothesized segmentation with each correct segmentation, as is standardly done in MorphoChallenge.

6.1 UMS Experiments

We first evaluate CHIPMUNK on UMS, by predicting LMS and then discarding the labels. Our primary baseline is the state-of-the-art supervised system CRF-MORPH of Ruokolainen et al. (2013). We ran the version of the system that the authors published on their website.[8] We optimized the model's two hyperparameters on Tune: the number

[5] http://research.ics.aalto.fi/events/morphochallenge2010/

[6] https://github.com/desmond86/Indonesian-English-Bilingual-Corpus

[7] We used both Tune and Dev in order to both optimize hyperparameters on held-out data (Tune) and perform qualitative error analysis on separate held-out data (Dev).

[8] http://users.ics.tkk.fi/tpruokol/software/crfs_morph.zip

	English	Finnish	Indonesian	German	Turkish	Zulu
CRF-MORPH	83.23	81.98	93.09	84.94	88.32	88.48
CRF-MORPH +LSV	84.45	84.35	93.50	86.90	89.98	89.06
First-order CRF	84.66	85.05	93.31	85.47	90.03	88.99
Higher-order CRF	84.66	84.78	93.88	85.40	90.65	88.85
CHIPMUNK	84.40	84.40	93.76	85.53	89.72	87.80
CHIPMUNK +Morph	83.27	84.71	93.17	84.84	90.48	90.03
CHIPMUNK +Affix	83.81	86.02	93.51	85.81	89.72	89.64
CHIPMUNK +Dict	86.10	86.11	95.39	87.76	90.45	88.66
CHIPMUNK +Dict,+Affix,+Morph	**86.31**	**88.38**	**95.41**	**87.85**	**91.36**	**90.16**

Table 5: Test F_1 for UMS. Features: LSV = letter successor variety, Affix = affix, Dict = dictionary, Morph = optimal (on Tune) morphotactic tagset.

of epochs and the maximal length of n-gram character features. The system also supports Harris's letter successor variety (LSV) features (Section 5), extracted from large unannotated corpora, our second baseline. For completeness, we also compare CHIPMUNK with a first-order CRF and a higher-order CRF (Müller et al., 2013), both used the same n-gram features as CRF-MORPH, but without the LSV features.[9] We evaluate all models using the traditional macro F_1 of the segmentation boundaries.

Discussion. The UMS results on held-out data are displayed in Table 5. Our most complex model beats the best baseline by between 1 (German) and 3 (Finnish) points F1 on all six languages. We additionally provide extensive ablation studies to highlight the contribution of our novel features. We find that the properties of each specific language highly influences which features are most effective. For the agglutinative languages, i.e, Finnish, Turkish and Zulu, the affix based features (+Affix) and the morphotactic tagset (+Morph) yield consistent improvements over the semi-CRF models with a single state. Improvements for the affix features range from 0.2 for Turkish to 2.14 for Zulu. The morphological tagset yields improvements of 0.77 for Finnish, 1.89 for Turkish and 2.10 for Zulu. We optimized tagset granularity on Tune and found that levels 4 and level 2 yielded the best results for the three agglutinative and the three other languages, respectively.

The dictionary features (+Dict) help universally, but their effects are particularly salient in languages with productive compounding, i.e., English, Finnish and German, where we see improvements of > 1.7.

[9]Model order, maximal character n-gram length and regularization coefficients were optimized on Tune.

		+Affix	+Dict,+Affix
Level 0	90.11	90.13	91.66
Level 1	90.73	90.68	92.80
Level 2	89.80	90.46	92.04
Level 3	91.03	90.83	92.31
Level 4	91.80	92.19	**93.21**

Table 6: Example of the effect of larger tagsets (Figure 2) on Turkish segmentation measured on our development set. As Turkish is an agglutinative language with hundreds of affixes, the efficacy of our approach is expected to be particularly salient here. Recall we optimized for the best tagset granularity for our experiments on Tune.

In comparison with previous work (Ruokolainen et al., 2013) we find that our most complex model yields consistent improvements over CRF-MORPH +LSV for all languages: The improvements range from > 1 for German over > 1.5 for Zulu, English, and Indonesian to > 2 for Turkish and > 4 for Finnish.

To illustrate the effect of modeling morphotactics through the larger morphotactic tagset on performance, we provide a detailed analysis of Turkish. See Table 6. We consider three different feature sets and increase the size of the morphotactic tagsets depicted in Figure 2. The results evince the general trend that improved morphotactic modeling benefits segmentation. Additionally, we observe that the improvements are complementary to those from the other features.

As discussed earlier, a key problem in UMS, especially in low-resource settings, is the detection of novel roots and affixes. Since many of our features were designed to combat this problem specifically, we investigated this aspect independently. Table 7 shows the number of novel roots and affixes found by our best model and the baseline. In all languages, CHIPMUNK correctly identifies between

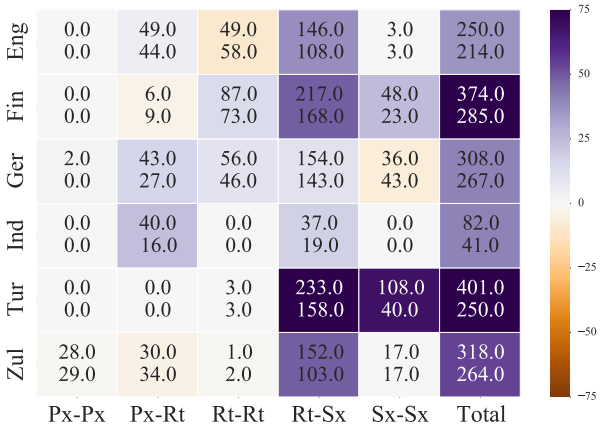

Figure 3: This figure represents a comparative analysis of undersegmentation. Each column (labels at the bottom) shows how often CRF-MORPH +LSV (top number in heatmap) and CHIPMUNK (bottom number in heatmap) select a segment that is two separate segments in the gold standard. E.g., Rt-Sx indicates how a root and a suffix were treated as a single segment. The color depends on the difference of the two counts.

5% (English) and 22% (Finnish) more novel roots than the baseline. We do not see major improvements for affixes, but this is of less interest as there are far fewer novel affixes.

We further explore how CHIPMUNK and the baseline perform on different boundary types by looking at missing boundaries between different morphotactic types; this error type is also known as *undersegmentation*. Figure 3 shows a heatmap that overviews errors broken down by morphotactic tag. We see that most errors are caused between root and suffixes across all languages. This is related to the problem of finding new roots, as a new root is often mistaken as a root-affix composition.

6.2 Root Detection and Stemming

Root detection[1] and stemming[1] are two important NLP problems that are closely related to morphological segmentation and used in applications such as MT, information retrieval, parsing and information extraction. Here we explore the utility of CHIPMUNK as a statistical stemmer and root detector.

Stemming is closely related to the task of *lemmatization*, which involves the additional step of normalizing to the canonical form.[10] Consider the German particle verb participle *auf-ge-schrieb-en* 'written down'. The participle is built by applying an alternation to the verbal root *schreib* 'write' adding the participial circumfix *ge-en* and finally adding the verb particle *auf*. In our segmentation-based definition, we would consider *schrieb* 'write' as its root and *auf-schrieb* as its stem. In order to additionally to restore the lemma, we would also have to reverse the stem alternation that replaced *ei* with *ie* and add the infinitival ending *en* yielding the infinitive *auf-schreib-en*.

[10]Thus in our experiments there are *no* stem alternations. The output is equivalent to that of the Porter stemmer (Porter, 1980).

	CRF-MORPH		CHIPMUNK	
	Roots	Affixes	Roots	Affixes
English	614	6	644	12
Finnish	502	10	613	11
German	360	6	414	9
Indonesian	593	0	639	0
Turkish	435	22	514	19
Zulu	146	10	160	11

Table 7: Dev number of unseen root and affix types correctly identified by CRF-MORPH +LSV and CHIPMUNK +Affix,+Dict,+Morph.

Our baseline MORFETTE (Chrupała et al., 2008) is a statistical transducer that first extracts edit paths between input and output and then uses a perceptron classifier to decide which edit path to apply. In short, MORFETTE treats the task as a string-to-string transduction problem, whereas we view it as a labeled segmentation problem.[11] Note, that MORFETTE would in principle be able to handle stem alternations, although these usually lead to an increase in the number of edit paths. We use level 2 tagsets for all experiments—the smallest tagsets complex enough for stemming—and extract the relevant segments.

Discussion. Our results are shown in Table 8. We see consistent improvements across all tasks. For the fusional languages (English, German and Indonesian) we see modest gains in performance on both root detection and stemming. However, for the agglutinative languages (Finnish, Turkish and Zulu) we see absolute gains as high as 50% (Turkish) in accuracy. This significant improvement is due to the complexity of the tasks in these languages—their productive morphology increases sparsity and makes the unstructured string-to-string transduction approach suboptimal. We view this as solid evidence that labeled segmentation has utility in many components of the NLP pipeline.

[11]Note that MORFETTE is a pipeline that first tags and *then* lemmatizes. We only make use of this second part of MORFETTE for which it is a strong string-to-string transduction baseline.

		English	Finnish	German	Indonesian	Turkish	Zulu
Root Detection	MORFETTE	62.82	39.28	43.81	86.00	26.08	30.76
	CHIPMUNK	**70.31**	**69.85**	**67.37**	**90.00**	**75.62**	**62.23**
Stemming	MORFETTE	91.35	51.74	79.49	86.00	28.57	58.12
	CHIPMUNK	**94.24**	**79.23**	**85.75**	**89.36**	**85.06**	**67.64**

Table 8: Test Accuracies for root detection and stemming.

		Finnish	Turkish
F1	MaxEnt	75.61	69.92
	MaxEnt +Split	74.02	76.61
	CHIPMUNK +All	**80.34**	**85.07**
Acc.	MaxEnt	60.96	37.88
	MaxEnt +Split	59.04	44.30
	CHIPMUNK +All	**65.00**	**56.06**

Table 9: Test F-Scores / accuracies for morphological tag classification.

	Morpheme Tags	Full Word Tags
Finnish	43	172
Turkish	50	636

Table 10: Number of full word and morpheme tags in the datasets.

6.3 Morphological Tag Classification

The joint modeling of segmentation and morphotactic tags allows us to use CHIPMUNK for a crude form of morphological analysis: the task of *morphological tag classification*, which we define as *annotation of a word with its most likely inflectional features*.[12] To be concrete, our task is to predict the inflectional features of word type based only on its character sequence and not its sentential context. To this end, we take Finnish and Turkish as two examples of languages that should suit our approach particularly well as both have highly complex inflectional morphologies. We use our most fine-grained tagset and replace all non-inflectional tags with a simple segment tag. The tagset sizes are listed in Table 10.

We use the same experimental setup as in Section 6.2 and compare CHIPMUNK to a maximum entropy classifier (MaxEnt), whose features are character n-grams of up to a maximal length of k. [13] The maximum entropy classifier is L_1-regularized and its regularization coefficient as well as the value for k are optimized on Tune. As a second, stronger baseline we use a MaxEnt classifier that splits tags into their constituents and concatenates the features with every constituent as well as the complete tag (MaxEnt +Split). Both of the baselines in Table 9 are 0th-order versions of the state-of-the-art CRF-based morphological tagger MARMOT (Müller et al., 2013) (since our model is type-based), making this a strong baseline. We report full analysis accuracy and macro F_1 on the set of individual inflectional features.

Discussion. The results in Table 9 show that our proposed method outperforms both baselines on both performance metrics. We see gains of over 6% in accuracy in both languages. This is evidence that our proposed approach could be successfully integrated into a morphological tagger to give a stronger character-based signal.

7 Comparison to Finite-State Morphology

A morphological finite-state analyzer is customarily a hand-crafted tool that generates all the possible morphological readings with their associated features. We believe that, for many applications, high quality finite-state morphological analysis is superior to our techniques. Finite-state morphological analyzers output a small set of linguistically valid analyses of a type, typically with only limited overgeneration. However, there are two significant problems. The first is that significant effort is required to develop the transducers modeling the "grammar" of the morphology and there is significant effort in creating and updating the lexicon. The second is, it is difficult to use finite-state morphology to guess analyses involving roots not covered in the lexicon.[14] In fact, this is usually solved by viewing it as a different problem, *morphological guessing*, where linguistic knowledge similar to the features we have presented is used to try to guess POS and morphological analysis for types with no analysis.

[12] We recognize that this task is best performed with sentential context (token-based). Integration with a POS tagger, however, is beyond the scope of this paper.

[13] Prefixes and suffixes are explicitly marked.

[14] While one can in theory put in wildcard root states, this does not work in practice due to overgeneration.

In contrast, our training procedure learns a probabilistic transducer, which is a soft version of the type of hand-engineered grammar that is used in finite-state analyzers. The 1-best labeled morphological segmentation our model produces offers a simple and clean representation which will be of great use in many downstream applications. Furthermore our model unifies analysis and guessing into a single simple framework. Nevertheless, finite-state morphologies are still extremely useful, high-precision tools. A primary goal of future work will be to use CHIPMUNK to attempt to induce higher-quality morphological processing systems.

8 Conclusion and Future Work

We have presented *labeled morphological segmentation* (LMS) in this paper, a new approach to morphological processing. LMS unifies three tasks that were solved before by different methods—unlabeled morphological segmentation, stemming, and morphological tag classification. LMS annotation itself has great potential for use in downstream NLP applications. Our hierarchy of labeled morphological segmentation tagsets can be used to map the heterogeneous data in six languages we work with to universal representations of different granularities. We plan future creation of gold standard segmentations in more languages using our annotation scheme.

We further presented CHIPMUNK a semi-CRF-based model for LMS that allows for the integration of various linguistic features and consistently out-performs previously presented approaches to *unlabeled morphological segmentation*. An important extension of CHIPMUNK is embedding it in a context-sensitive POS tagger. Current state-of-the-art models only employ character level n-gram features to model word-internals (Müller et al., 2013). We have demonstrated that our structured approach outperforms this baseline. We leave this natural extension to future work.

The datasets used in this work, additional description of our novel tagsets and CHIPMUNK can be found at http://cistern.cis.lmu.de/chipmunk.

Acknowledgments

We would like to thank Jason Eisner, Helmut Schmid, Özlem Çetinoğlu and the anonymous reviewers for their comments. This material is based upon work supported by a Fulbright fellowship awarded to the first author by the German-American Fulbright Commission and the National Science Foundation under Grant No. 1423276. The second author is a recipient of the Google Europe Fellowship in Natural Language Processing, and this research is supported by this Google Fellowship. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644402 (HimL) and the DFG grant *Models of Morphosyntax for Statistical Machine Translation*.

References

Muhammad Abdul-Mageed, Mona T. Diab, and Sandra Kübler. 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language*.

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of EMNLP*.

R Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1993. The CELEX lexical database on CD-ROM. Technical report, Linguistic Data Consortium.

Antal Van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of ACL*. Association for Computational Linguistics.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*.

Stanley F. Chen and Ronald Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical report, DTIC Document.

Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *Proceedings of LREC*.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraclar, and Andreas Stolcke. 2007a. Analysis of morph-based speech recognition and the modeling of out-of-vocabulary words across languages. In *Proceedings of HLT-NAACL*.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti

Varjokallio, Ebru Arisoy, Murat Saraclar, and Andreas Stolcke. 2007b. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *TSLP*.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of SIGMORPHON*.

Mathias Creutz and Krista Lagus. 2004. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of SIGMORPHON*.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL*.

Elif Eyigöz, Daniel Gildea, and Kemal Oflazer. 2013. Simultaneous word-morpheme alignment for statistical machine translation. In *Proceedings of HLT-NAACL*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING*.

Zellig Harris. 1995. From phoneme to morpheme. *Language*.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010a. Semi-supervised learning of concatenative morphology. In *Proceedings of SIGMORPHON*.

Oskar Kohonen, Sami Virpioja, Laura Leppänen, and Krista Lagus. 2010b. Semi-supervised extensions to Morfessor baseline. In *Proceedings of the Morpho Challenge Workshop*.

Lucia D. Krisnawati and Klaus U. Schulz. 2013. Plagiarism detection for Indonesian texts. In *Proceedings of iiWAS*.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge competition 2005–2010: Evaluations and results. In *Proceedings of SIGMORPHON*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Septina Dian Larasati, Vladislav Kuboň, and Daniel Zeman. 2011. Indonesian morphology tool (morphind): Towards an Indonesian corpus. In *Systems and Frameworks for Computational Morphology*. Springer.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*.

Erwin Marsi, Antal van den Bosch, and Abdelhadi Soudi. 2005. Memory-based morphological analysis generation and part-of-speech tagging of Arabic. In *Proceedings of ACL Workshop: Computational Approaches to Semitic Languages*.

Kevin P Murphy. 2002. Hidden semi-Markov models (hsmms). Technical report, Massachusetts Institute of Technology.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of EMNLP*.

Karthik Narasimhan, Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *Proceedings of EMNLP*.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of EMNLP*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of NAACL*.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

Lawrence Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of CoNLL*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. *Proceedings of EACL*.

Sunita Sarawagi and William W Cohen. 2004. Semi-Markov conditional random fields for information extraction. In *Proceedings of NIPS*.

Andrew Smith and Miles Osborne. 2006. Using gazetteers in discriminative information extraction. In *Proceedings of CoNLL*.

Sebastian Spiegler, Andrew Van Der Spuy, and Peter A Flach. 2010. Ukwabelana: An open-source morphological Zulu corpus. In *Proceedings of COLING*.

Sami Virpioja, Oskar Kohonen, and Krista Lagus. 2010. Unsupervised morpheme analysis with Allomorfessor. In *Multilingual Information Access Evaluation*. Springer.

Martin J Wainwright and Michael I Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL*.

A Semi-CRF

Semi-CRFs (Sarawagi and Cohen, 2004) generalize linear-chain CRFs and model segmentation in addition to sequence labeling. Linear-chain CRFs are discriminative adaptations of *hidden Markov Models* (Lafferty et al., 2001); semi-CRFs are an analogous adaptation of *hidden semi-Markov models* (Murphy, 2002). The major advantage of semi-CRFs is that we can have features which look at complete segments, which allows us to elegantly integrate our new features.

A semi-CRF represents $\boldsymbol{w}$ (a word) as a sequence of segments $\boldsymbol{s} = \langle s_1, \ldots, s_n \rangle$, each of which is assigned a label ℓ_i. The segments s_i are required to have positive length. The concatenation of all segments equals the word $\boldsymbol{w}$. We seek a log-linear distribution $p_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\ell} \mid \boldsymbol{w})$ over all possible segmentations and label sequences for word $\boldsymbol{w}$ where $\boldsymbol{f}$ is the feature function and $\boldsymbol{\theta}$ is the parameter vector. Note we recover the standard CRF if we restrict segment length to 1. Formally, we define the distribution as

$$p_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{\ell} \mid \boldsymbol{w}) \stackrel{\text{def}}{=} \frac{1}{Z_{\boldsymbol{\theta}}(\boldsymbol{w})} \prod_{i=1}^{|\boldsymbol{s}|} e^{\boldsymbol{\theta}^T \boldsymbol{f}(\boldsymbol{w}, s_i, \ell_i, \ell_{i-1}, i)},$$

where $\boldsymbol{f}$ is the feature function and $Z_{\boldsymbol{\theta}}(\boldsymbol{w})$

$$Z_{\boldsymbol{\theta}}(\boldsymbol{w}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{s}, \boldsymbol{\ell}} \prod_{i=1}^{|\boldsymbol{s}|} e^{\boldsymbol{\theta}^T \boldsymbol{f}(\boldsymbol{w}, s_i, \ell_i, \ell_{i-1}, i)},$$

is the partition function, which ensures that the measure is normalized. We use a generalization of the forward-backward algorithm for efficient computation of the gradient (Sarawagi and Cohen, 2004).

The dynamic programming recursions are quite similar to those used for a standard CRF and HMM (Rabiner, 1989). Inspection of the semi-Markov forward recursion,

$$\boldsymbol{\alpha}(t, \ell) = \sum_{i=1}^{t-1} \sum_{\ell'=1}^{L} e^{\boldsymbol{\theta}^T \boldsymbol{f}(\boldsymbol{w}, w_{t-i,t}, \ell, \ell', i)} \cdot \boldsymbol{\alpha}(t-i, \ell'),$$

shows that the program runs in $\mathcal{O}(n^2 \cdot L^2)$ time where n is the length of the word $\boldsymbol{w}$ and L is the number of labels (size of the tagset). While only a factor of n slower than a standard linear-chain CRF, the semi-CRF can be significantly slower in practice due to the frequency of long words (more than 20 letters) in agglutinative languages such as Turkish and Finnish.[15]

A.1 Parameter Estimation

All models were optimized using the L-BFGS algorithm — a general purpose algorithm for moderate-scale non-linear gradient-based optimization (Liu and Nocedal, 1989). We combat overfitting with a simple $\lambda \cdot ||\cdot||_2^2$ regularizer on the parameters. This acts as a Gaussian prior over the weights and disfavors single, large parameter values by encouraging various weights to "share the load". As in all exponential family models, the gradient of the log-likelihood takes the form of the difference between the observed feature counts and expected feature counts (Wainwright and Jordan, 2008). In the case of the semi-CRF, we have the following expression:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{|\boldsymbol{s}|} \boldsymbol{f}(\boldsymbol{w}, s_i, \ell_i, \ell_{i-1}) -$$

$$\sum_{\boldsymbol{s}', \boldsymbol{\ell}'} \sum_{i=1}^{|\boldsymbol{s}|} p_{\boldsymbol{\theta}}(s'_i, \ell'_i \mid \ell'_{i-1}, \boldsymbol{w}) \cdot \boldsymbol{f}(\boldsymbol{w}, s'_i, \ell'_i, \ell'_{i-1}),$$

where $\mathcal{L}$ is the log-likelihood, $\boldsymbol{s}$ is the observed sequence of segments, $\boldsymbol{\ell}$ is the observed sequence of labels and $\boldsymbol{w}$ is the word. Note the expected counts are also computed using the forward-backward algorithm.[16] (Chen and Rosenfeld, 1999).

B Additional Tables

English	119,839
Finnish	6,690,417
German	364,564
Indonesian	35,269
Turkish	80,261
Zulu	73,525

Table 11: Number of words covered by the respective ASPELL dictionary

[15] This recursion describes an algorithm that allows for sequences with arbitrary length. If we have a maximum segment limit of m, we get an algorithm that runs in $\mathcal{O}(n \cdot m \cdot L^2)$ time as in (Sarawagi and Cohen, 2004).

[16] This is the gradient for a *single* word–we have omitted the sum over the data for simplicity.

	Morpheme Tags	Full Word Tags
Finnish	43	172
Turkish	50	636

Table 12: Number of full word and morpheme tags in the datasets.