Morphological Segmentation Inside-Out

Ryan Cotterell
Department of Computer Science
Johns Hopkins University
ryan.cotterell@jhu.edu

Arun Kumar
Faculty of Arts and Humanities
Universitat Oberta de Catalunya

Hinrich Schütze
CIS
LMU Munich

Abstract
Morphological segmentation has traditionally been modeled with non-hierarchical models, which yield flat segmentations as output. In many cases, however, proper morphological analysis requires hierarchical structure—especially in the case of derivational morphology. In this work, we introduce a discriminative joint model of morphological segmentation along with the orthographic changes that occur during word formation. To the best of our knowledge, this is the first attempt to approach discriminative segmentation with a context-free model. Additionally, we release an annotated treebank of 7454 English words with constituency parses, encouraging future research in this area.

1 Introduction
In NLP, supervised morphological segmentation has typically been viewed as either a sequence labeling or a segmentation task (Ruokolainen et al., 2016). In contrast, we consider a hierarchical approach, employing a context-free grammar (CFG). CFGs provide a richer model of morphology: they capture (i) the intuition that words themselves have internal constituents, which belong to different categories, as well as (ii) the order in which affixes are attached. Moreover, many morphological processes, e.g., compounding and reduplication, are best modeled as hierarchical; thus, context-free models are expressively more appropriate.

The purpose of morphological segmentation is to decompose words into smaller units, known as morphemes, which are typically taken to be the smallest meaning bearing units in language. This work concerns itself with modeling hierarchical structure over these morphemes. Note a simple flat morphological segmentation can also be straightforwardly derived from the CFG parse tree. Segmentations have found use in a diverse set of NLP applications, e.g., automatic speech recognition (Afify et al., 2006), keyword spotting (Narasimhan et al., 2014), machine translation (Clifton and Sarkar, 2011) and parsing (Seeker and Çetinoğlu, 2015). In contrast to prior work, we focus on canonical segmentation, i.e., we seek to jointly model orthographic changes and segmentation. For instance, the canonical segmentation of untestably is un+test+able+ly, where we map ably to able+ly, restoring the letters le.

We make two contributions: (i) We introduce a joint model for canonical segmentation with a CFG backbone. We experimentally show that this model outperforms a semi-Markov model on flat segmentation. (ii) We release the first morphology treebank, consisting of 7454 English word types, each annotated with a full constituency parse.

2 The Case For Hierarchical Structure
Why should we analyze morphology hierarchically? It is true that we can model much of morphology with finite-state machinery (Beesley and Karttunen, 2003), but there are, nevertheless, many cases where hierarchical structure appears requisite. For instance, the flat segmentation of the word untestably→un+test+able+ly is missing important information about how the word was derived. The correct parse [[un][testable][ly]], on the other hand, does tell us that this is the order in which the com-
plex form was derived:

\[
testable \rightarrow test \rightarrow un \rightarrow untestable \rightarrow untestably.
\]

This gives us clear insight into the structure of the lexicon—we should expect that the segment testable exists as an independent word, but ably does not.

Moreover, a flat segmentation is often semantically ambiguous. There are two potentially valid readings of untestably depending on how the negation prefix un scopes. The correct tree (see Figure 1) yields the reading “in the manner of not able to be tested”. A second—likely infelicitous—reading where the segment untest forms a constituent yields the reading “in a manner of being able to untest”. Recovering the hierarchical structure allows us to select the correct reading; note there are even cases of true ambiguity; e.g., unlockable has two readings: “unable to be locked” and “able to be unlocked”.

Figure 1: Canonical segmentation parse trees for untestably and unlockable. For both words, the scope of un is ambiguous. Arguably, (a) is the only correct parse tree for untestably; the reading associated with (b) is hard to get. On the other hand, unlockable is truly ambiguous between “able to be unlocked” (c) and “unable to be locked” (d).

3 Parsing the Lexicon

A novel component of this work is the development of a discriminative parser (Finkel et al., 2008; Hall et al., 2014) for morphology. The goal is to define a probability distribution over all trees that could arise from the input word, after reversal of orthographic and phonological processes. We employ the simple grammar shown in Table 1. Despite its simplicity, it models the order in which morphemes are attached.

More formally, our goal is to map a surface form \( w \) (e.g., \( w=\text{untestably} \)) into its underlying canonical form \( u \) (e.g., \( u=\text{untestably} \)) and then into a parse tree \( t \) over its morphemes. We assume \( u, w \in \Sigma^* \), for some discrete alphabet \( \Sigma \).

Note that a parse tree over the string implicitly defines a flat segmentation given our grammar—one can simply extract the characters spanned by all preterminals in the resulting tree. Before describing the joint model in detail, we first consider its pieces individually.

3.1 Restoring Orthographic Changes

To extract a canonical segmentation (Naradowsky and Goldwater, 2009; Cotterell et al., 2016), we restore orthographic changes that occur during word formation. To this end, we define the score function,

\[
\text{score}_\eta(u, w) = \sum_{a \in A(u, w)} \exp \left( g(u, a, w) \top \eta \right),
\]

where \( A(u, w) \) is the set of all monotonic alignments between the strings \( u \) and \( w \). The goal is for \( \text{score}_\eta \) to assign higher values to better matched pairs, e.g., \( (w=\text{untestably}, u=\text{untestably}) \).

We have left these out of the equation for simplicity, but we refer to the reader Dreyer et al. (2008) for a more thorough exposition. To ensure that the partition function \( Z_\eta(w) \) is finite, we cap the maximum string length \( u \), which yields

\[
Z_\eta(w) = \sum_{u' \in \Sigma^{|w|+k}} \exp \left( g(u', w) \top \eta \right),
\]

\(^1\)For efficiency, we assume \( u \in \Sigma^{|w|+k}, k = 5.\)
Next, we need to score an underlying canonical form (e.g., \textit{un-testable-ly}) together with a parse tree (e.g., \textit{t=\{[\text{in}][\text{test}][\text{able}][\text{ly}]\}}). Thus, we define the parser score

\[
\text{score}_\omega(t, u) = \exp \left( \sum_{\pi \in \Pi(t)} f(\pi, u)^\top \omega \right),
\]

where \(\Pi(t)\) is the set of \textit{anchored productions} in the tree \(t\). An anchored production \(\pi\) is a grammar rule in Chomsky normal form attached to a span, e.g., \(A_{i,k} \rightarrow B_{i,j}C_{j+1,k}\). Each \(\pi\) is then assigned a weight by the linear function \(f(\pi, u)^\top \omega\), where the function \(f\) extracts relevant features from the anchored production as well as the corresponding span of the underlying form \(u\). This model is typically referred to as a weighted CFG (WCFG) (Smith and Johnson, 2007) or a CRF parser.

Luckily, we can exploit the structure of the problem to efficiently compute the partition function,

\[
Z_\omega(u) = \sum_{t \in \mathcal{T}(u)} \exp \left( \sum_{\pi \in \Pi(t)} f(\pi, u)^\top \omega \right),
\]

where \(\mathcal{T}(u)\) is the set of all trees under the grammar that have yield \(u\). Specifically, we make use of the inside algorithm, which is just CKY (Aho and Ullman, 1979) in the \((+, \times)\) semiring (Goodman, 1998), which runs in \(O(|G| \cdot |u|^2)\) time, where \(|G|\) is the size of the grammar.

For \(f\), we define three span features: (i) indicator features on the span’s segment, (ii) an indicator feature that fires if the segment appears in an external corpus\(^2\) and (iii) the conjunction of the segment with the label (e.g., \textsc{prefix}) of the subtree root. Following Hall et al. (2014), we employ an indicator feature for each production as well as production backoff features.

\[\text{\textsc{3.2\ Morphological Analysis as Parsing}}\]

Our complete model is a joint CRF (Koller and Friedman, 2009) where each of the above scores are factors. We define the likelihood as

\[
p_\theta(t, u \mid w) = \frac{1}{Z_\theta} \text{score}_\omega(t, u) \cdot \text{score}_\eta(u, w),
\]

where \(\theta = \{\omega, \eta\}\) is the parameter vector and

\[
Z_\theta = \sum_{u' \in \Sigma^*} \sum_{t' \in \mathcal{T}_{u'}} \text{score}_\omega(t', u') \cdot \text{score}_\eta(u', w)
\]

is the partition function and \(\mathcal{T}_{u'}\) is set of all parse trees for the string \(u'\). We see now that both WFST and WCFG are just structured factors in the model.

\[\text{\textsc{4\ A Joint Model}}\]

---

\(^2\)We use the Wikipedia dump from 2016-05-01.
The joint approach has the advantage that it allows both factors to work together to influence the choice of the underlying form $u$. This is useful as the parser now has access to which words are attested in the language; this helps guide the relatively weak transduction model. On the downside, the partition function $Z_\theta$ now involves a sum over both all strings in $\sum_{|w|+k}$ and all possible parses of each string! Inference in this joint model is intractable, so we resort to approximate methods.

4.1 Learning and Inference

We use stochastic gradient descent to optimize the log-probability of the training data $\sum_{i=1}^N \log p_\theta(t^{(i)}, u^{(i)} | w^{(i)})$; this requires the computation of the gradient of the partition function $\nabla_\theta \log Z_\theta$, which is intractable. As in all CRFs, this gradient is in fact an expectation:

$$\nabla_\theta \log Z_\theta = \mathbb{E}_{(u,t) \sim p_\theta} \left[ \log \text{score}_\omega(t, u) \cdot \text{score}_\eta(u, w) \right].$$

To approximate this expectation, we use an importance sampling routine. Roughly speaking, we approximate the hard-to-sample-from distribution $p_\theta$ by taking samples from an easy-to-sample-from proposal distribution $q$. We then reweight the samples using the unnormalized score from $p_\theta$. Due to a lack of space, we omit the derivation of the approximate gradient.

4.2 Decoding

We also decode by importance sampling. Given $w$, we sample canonical forms $u$ and then run the CKY algorithm to get the highest scoring tree.

5 Related Work

We believe our attempt to train discriminative grammars for morphology is novel. Nevertheless, other researchers have described parsers for morphology. Most of this work is unsupervised: Johnson et al. (2007) applied a Bayesian PCFG to unsupervised morphological segmentation. Similarly, Adaptor Grammars (Johnson et al., 2006), a non-parametric Bayesian generalization of PCFGs, have been applied to the unsupervised version of the task (Botha and Blunsom, 2013; Sirtis and Goldwater, 2013). Relatedly, Schmid (2005) performed unsupervised disambiguation of a German morphological analyzer (Schmid et al., 2004) using a PCFG, using the inside-outside algorithm (Baker, 1979). Also, discriminative parsing approaches have been applied to the related problem of Chinese word segmentation (Zhang et al., 2014).

6 Morphological Treebank

Supervised morphological segmentation has historically been treated as a segmentation problem, devoid of hierarchical structure. A core reason behind this is that—to the best of our knowledge—there are no hierarchically annotated corpora for the task. To remedy this, we provide tree annotations for a subset of the English portion of CELEX (Baayen et al., 1993). We reannotated 7454 English types with a full constituency parse.\footnote{In many cases, we corrected the flat segmentation as well.} The resource will be freely available for future research.

6.1 Annotation Guidelines

The annotation of the morphology treebank was guided by three core principles. The first principle concerns productivity: we exclusively annotate productive morphology. In the context of morphology, productivity refers to the degree that native speakers actively employ the affix to create new words (Aronoff, 1976). We believe that for NLP applications, we should focus on productive affixation. Indeed, this sets our corpus apart from many existing morphologically annotated corpora such as CELEX. For example, CELEX contains warmth$\rightarrow$warm$+$th, but th is not a productive suffix and cannot be used to create new words. Thus, we do not want to analyze hearth$\rightarrow$hear$+$th or, in general, allow wug$\rightarrow$wug$+$th. Second, we annotate for semantic coherence. When there are several candidate parses, we choose the one that is best compatible with the compositional semantics of the derived form.

Interestingly, multiple trees can be considered valid depending on the linguistic tier of interest. Consider the word unhappier. From a semantic perspective, we have the parse $[[un [happy]] \text{ er}]$ which gives us the correct meaning “not happy to a greater degree”. However, since the suffix er only attaches to mono- and bisyllabic words, we get $[un[[happy] \text{ er}]]$ from a phonological perspective. In the linguis-
tics literature, this problem is known as the bracketing paradox (Pesetsky, 1985; Embick, 2015). We annotate exclusively at the syntactic-semantic tier.

Thirdly, in the context of derivational morphology, we force spans to be words themselves. Since derivational morphology—by definition—forms new words from existing words (Lieber and Štekauer, 2014), it follows that each span rooted with \textsc{Word} or \textsc{Root} in the correct parse corresponds to a word in the lexicon. For example, consider \textit{unlickable}. The correct parse, under our scheme, is \texttt{[un [lick] able]}. Each of the spans (\textit{lick}, \textit{lickable} and \textit{unlickable}) exists as a word. By contrast, the parse \texttt{[un [lick] able]} contains the span \textit{unlick}, which is not a word in the lexicon. The span in the segmented form may involve changes, e.g., \texttt{[un [achieve] able]}, where \textit{achieveable} is not a word, but \textit{achievable} (after deleting \textit{e}) is.

### 7 Experiments

We run a simple experiment to show the empirical utility of parsing words—we compare a WCFG-based canonical segmenter with the semi-Markov segmenter introduced in Cotterell et al. (2016). We divide the corpus into 10 distinct train/dev/test splits with 5454 words for train and 1000 for each of dev and test. We report three evaluation metrics: full form accuracy, morpheme $F_1$ (Van den Bosch and Daelemans, 1999) and average edit distance to the gold segmentation with boundaries marked by a distinguished symbol. For the WCFG model, we also report constituent $F_1$—typical for sentential constituency parsing—as a baseline for future systems. This $F_1$ measures how well we predict the whole tree (not just a segmentation). For all models, we use $L_2$ regularization and run 100 epochs of \textsc{AdaGrad} (Duchi et al., 2011) with early stopping. We tune the regularization coefficient by grid search considering $\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

<table>
<thead>
<tr>
<th></th>
<th>Segmentation</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Morph. $F_1$</td>
<td>Edit Acc.</td>
</tr>
<tr>
<td>Flat</td>
<td>78.89 (0.9)</td>
<td>0.72 (0.04)</td>
</tr>
<tr>
<td>Hier</td>
<td>85.55 (0.6)</td>
<td>0.55 (0.03)</td>
</tr>
</tbody>
</table>

Table 2: Results for the 10 splits of the treebank. Segmentation quality is measured by morpheme $F_1$, edit distance and accuracy; tree quality by constituent $F_1$.

### 7.1 Results and Discussion

Table 2 shows the results. The hierarchical WCFG model outperforms the flat semi-Markov model on all metrics on the segmentation task. This shows that modeling structure among the morphemes, indeed, does help segmentation. The largest improvements are found under the morpheme $F_1$ metric ($\approx 6.5$ points). In contrast, accuracy improves by $< 1\%$. Edit distance is in between with an improvement of 0.2 characters. Accuracy, in general, is an all or nothing metric since it requires getting every canonical segment correct. Morpheme $F_1$, on the other hand, gives us partial credit. Thus, what this shows us is that the WCFG gets a lot more of the morphemes in the held-out set correct, even if it only gets a few complete forms correct. We provide additional results evaluating the entire tree with constituency $F_1$ as a future baseline.

### 8 Conclusion

We presented a discriminative CFG-based model for canonical morphological segmentation and showed empirical improvements on its ability to segment words under three metrics. We argue that our hierarchical approach to modeling morphemes is more often appropriate than the traditional flat segmentation. Additionally, we have annotated 7454 words with a morphological constituency parse. The corpus is available online at \url{http://ryancotterell.github.io/data/morphological-treebank} to allow for exact comparison and to spark future research.

### Acknowledgements

The first author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship. The third author was supported by DFG (SCHU 2246/10-1).
References


Wolfgang Seeker and Özlem Çetinolu. 2015. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *TACL*.


