Morphological Segmentation Inside-Out

Ryan Cotterell[ə] **Arun Kumar**[ɣ] **Hinrich Schütze**[fi]
[ə]Department of Computer Science, Johns Hopkins University
[ɣ]Faculty of Arts and Humanities, Universitat Oberta de Catalunya
[fi]CIS, LMU Munich
{ryan.cotterell}@jhu.edu

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]

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*$\mapsto$*un+test+able+ly* is missing important information about how the word was derived. The correct *parse* [[*un*[[*test*]*able*]]*ly*], on the other hand, does tell us that this is the order in which the complex form was derived:

$$\textit{test} \xmapsto{\text{able}} \textit{testable} \xmapsto{\text{un}} \textit{untestable} \xmapsto{\text{ly}} \textit{untestably}.$$

This gives us insight into the structure of the lexicon—we 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

[1] *We found post publication that CELEX (Baayen et al., 1993) has annotated words for hierarchical morphological segmentation as well.*

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2325–2330
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

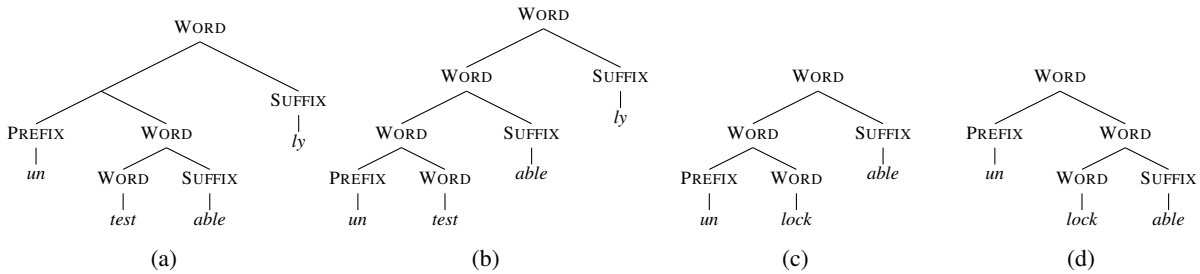

Figure 1: Canonical segmentation parse trees for *untestably* and *unlockable*. For both words, the scope of *un* is ambiguous. Arguably, (a) is the only correct parse tree for *untestably*; the reading associated with (b) is hard to get. On the other hand, *unlockable* is truly ambiguous between "able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

readings of *untestably* depending on how the negative 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."

We also note that theoretical linguists often implicitly assume a context-free treatment of word formation, e.g., by employing brackets to indicate different levels of affixation. Others have explicitly modeled word-internal structure with grammars (Selkirk, 1982; Marvin, 2002).

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=*untestably*) into its underlying canonical form u (e.g., u=*untestablely*) and then into a parse tree t over its morphemes. We assume $u, w \in \Sigma^*$, for some discrete alphabet Σ.[2] 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

$$score_{\boldsymbol{\eta}}(u, a, w) = \exp\left(\boldsymbol{g}(u, a, w)^\top \boldsymbol{\eta}\right) \quad (1)$$

where a is a monotonic alignment between the strings u and w. The goal is for score$_{\boldsymbol{\eta}}$ to assign higher values to better matched pairs, e.g., (w=*untestably*, u=*untestablely*). We refer to Dreyer et al. (2008) for a thorough exposition.

For ease of computation, we can encode this function as a weighted finite-state machine (WFST) (Mohri et al., 2002). This requires, however, that the feature function $\boldsymbol{g}$ factors over the topology of the finite-state encoding. Since our model conditions on the word w, the feature function $\boldsymbol{g}$ can extract features from *any* part of this string. Features on the output string, u, however, are more restricted. In this work, we employ a bigram model over output characters. This implies that each state remembers exactly one character: the previous one. See Cotterell et al. (2014) for details. We can compute the score for two strings u and w using a weighted generalization of the Levenshtein algorithm. Computing the partition function requires a different dynamic program, which runs in $\mathcal{O}(|w|^2 \cdot |\Sigma|^2)$ time. Note that since $|\Sigma| \approx 26$ (lower case English letters), it takes roughly $26^2 = 676$ times longer to compute the partition function than to score a pair of strings.

Our model includes several simple feature templates, including features that fire on individual edit actions as well as conjunctions of edit actions and

[2] For efficiency, we assume $u \in \Sigma^{|w|+k}$, $k = 5$.

ROOT	$\rightarrow$	WORD
WORD	$\rightarrow$	PREFIX WORD
WORD	$\rightarrow$	WORD SUFFIX
WORD	$\rightarrow$	Σ^+
PREFIX	$\rightarrow$	Σ^+
SUFFIX	$\rightarrow$	Σ^+

Table 1: The context-free grammar used in this work to model word formation. The productions closely resemble those of Johnson et al. (2006)'s Adaptor Grammar.

characters in the surrounding context. See Cotterell et al. (2016) for details.

3.2 Morphological Analysis as Parsing

Next, we need to score an underlying canonical form (e.g., u=*untestablely*) together with a parse tree (e.g., t=[[*un*[[*test*]*able*]]*ly*]). Thus, we define the parser score with the following function

$$score_{\boldsymbol{\omega}}(t, u) = \exp\left(\sum_{\pi \in \Pi(t)} \boldsymbol{f}(\pi, u)^\top \boldsymbol{\omega}\right) \quad (2)$$

where $\Pi(t)$ is the set of *anchored productions* in the tree t. An anchored production π is a grammar rule in Chomsky normal form attached to a span, e.g., $A_{i,k} \rightarrow B_{i,j} C_{j,k}$. Each π is then assigned a weight by the linear function $\boldsymbol{f}(\pi, u)^\top \boldsymbol{\omega}$, where the function $\boldsymbol{f}$ 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.

For $\boldsymbol{f}$, 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[3] and (iii) the conjunction of the segment with the label (e.g., PREFIX) of the subtree root. Following Hall et al. (2014), we employ an indicator feature for each production as well as production backoff features.

4 A Joint Model

Our complete model is a joint CRF (Koller and Friedman, 2009) where each of the above scores are factors. We define the following probability distribution over trees, canonical forms and their alignments to the original word

$$p_{\boldsymbol{\theta}}(t, a, u \mid w) = \frac{1}{Z_{\boldsymbol{\theta}}(w)} score_{\boldsymbol{\omega}}(t, u) \cdot score_{\boldsymbol{\eta}}(u, a, w) \quad (3)$$

where $\boldsymbol{\theta} = \{\boldsymbol{\omega}, \boldsymbol{\eta}\}$ is the parameter vector and the normalizing partition function as

$$Z_{\boldsymbol{\theta}}(w) = \sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u', w)} \sum_{t' \in \mathcal{T}(u')} score_{\boldsymbol{\omega}}(t', u') \cdot score_{\boldsymbol{\eta}}(u', a, w) \quad (4)$$

where $\mathcal{T}(u)$ is the set of all parse trees for the string u. This involves a sum over all possible underlying orthographic forms and all parse trees for those forms.

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 $Z_{\boldsymbol{\theta}}$ now involves a sum over all strings in $\Sigma^{|w|+k}$ *and* all possible parses of each string! Finally, we define the marginal distribution over trees and underlying forms as

$$p_{\boldsymbol{\theta}}(t, u \mid w) = \sum_{a \in A(u,w)} p_{\boldsymbol{\theta}}(t, a, u \mid w) \quad (5)$$

where $A(u, w)$ is the set of all monotonic alignments between u and w. The marginalized form in eq. (5) is our final model of morphological segmentation since we are not interested in the latent alignments a.

4.1 Learning and Inference

We use stochastic gradient descent to optimize the log-probability of the training data $\sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(t^{(n)}, u^{(n)} \mid w^{(n)})$; this requires the computation of the gradient of the partition function $\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}$. We may view this gradient as an expectation:

$$\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}(w) = \mathbb{E}_{(t,a,u) \sim p_{\boldsymbol{\theta}}(\cdot \mid w)} \left(\sum_{\pi \in \Pi(t)} \boldsymbol{f}(\pi, u)^\top + \boldsymbol{g}(u, a, w)^\top \right) \quad (6)$$

We provide the full derivation in Appendix A with an additional Rao-Blackwellization step that we

[3] We use the Wikipedia dump from 2016-05-01.

	Segmentation			Tree
	Morph. F_1	*Edit*	*Acc.*	*Const.* F_1
Flat	78.89 (0.9)	0.72 (0.04)	72.88 (1.21)	N/A
Hier	**85.55** (0.6)	**0.55** (0.03)	**73.19** (1.09)	79.01 (0.5)

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.

make use of in the implementation. While the sum over all underlying forms and trees in eq. (6) may be achieved in polynomial time (using the Bar-Hillel construction), we make use of an importance-sampling estimator, derived by Cotterell et al. (2016), which is faster in practice. Roughly speaking, we approximate the hard-to-sample-from distribution $p_{\boldsymbol{\theta}}$ by taking samples from an easy-to-sample-from proposal distribution q. Specifically, we employ a pipeline model for q consisting of WFST and then a WCFG sampled from consecutively. We then *reweight* the samples using the *unnormalized* score from $p_{\boldsymbol{\theta}}$. Importance sampling has found many uses in NLP ranging from language modeling (Bengio et al., 2003) and neural MT (Jean et al., 2015) to parsing (Dyer et al., 2016). Due to a lack of space, we omit the derivation of the importance-sampled 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; Sirts 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.[4] 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*↦*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*↦*hear*+*th* or, in general, allow *wug*↦*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*]] *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*] *er*]] from a phonological perspective. In the linguistics literature, this problem is known as the brack-

[4] In many cases, we corrected the flat segmentation as well.

eting 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 WORD or ROOT in the correct parse corresponds to a word in the lexicon. For example, consider *unlickable*. The correct parse, under our scheme, is [*un* [[*lick*] *able*]]. Each of the spans (*lick*, *lickable* and *unlickable*) exists as a word. By contrast, the parse [[*un* [*lick*]] *able*] contains the span *unlick*, which is not a word in the lexicon. The span in the segmented form may involve changes, e.g., [*un* [[*achieve*] *able*]], where *achieveable* is not a word, but *achievable* (after deleting *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 ADAGRAD (Duchi et al., 2011) with early stopping. We tune the regularization coefficient by grid search considering $\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

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 (≈ 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 more 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 `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

Mohamed Afify, Ruhi Sarikaya, Hong-Kwang Jeff Kuo, Laurent Besacier, and Yuqing Gao. 2006. On the use of morphological analysis for dialectal Arabic speech recognition. In *INTERSPEECH*.

Mark Aronoff. 1976. *Word Formation in Generative Grammar*. MIT Press.

R. Harald Baayen, Richard Piepenbrock, and Rijn van H. 1993. The CELEX lexical data base on CD-ROM.

James K. Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite-state Morphology: Xerox Tools and Techniques*. CSLI, Stanford.

Yoshua Bengio, Jean-Sébastien Senécal, et al. 2003. Quick training of probabilistic neural nets by importance sampling. In *AISTATS*.

Antal Van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *ACL*.

Jan A. Botha and Phil Blunsom. 2013. Adaptor grammars for learning non-concatenative morphology. In *EMNLP*, pages 345–356.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *ACL*.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2014. Stochastic contextual edit distance and probabilistic FSTs. In *ACL*.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *NAACL*.

Markus Dreyer, Jason R. Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *EMNLP*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *NAACL*.

David Embick. 2015. *The Morpheme: A Theoretical Introduction*, volume 31. Walter de Gruyter GmbH & Co KG.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *ACL*, volume 46, pages 959–967.

David Leo Wright Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *ACL*, pages 228–237.

Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *ACL*, pages 1–10.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *NIPS*, pages 641–648.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov Chain Monte Carlo. In *HLT-NAACL*, pages 139–146.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT press.

Rochelle Lieber and Pavol Štekauer. 2014. *The Oxford Handbook of Derivational Morphology*. Oxford University Press, USA.

Tatjana Marvin. 2002. *Topics in the Stress and Syntax of Words*. Ph.D. thesis, Massachusetts Institute of Technology.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Jason Naradowsky and Sharon Goldwater. 2009. Improving morphology induction by learning spelling rules. In *IJCAI*.

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

David Pesetsky. 1985. Morphology and logical form. *Linguistic Inquiry*, 16(2):193–246.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. Comparative study of minimally supervised morphological segmentation. *Computational Linguistics*.

Helmut Schmid. 2005. Disambiguation of morphological structure using a PCFG. In *EMNLP*, pages 515–522. Association for Computational Linguistics.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *LREC*.

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

Elisabeth Selkirk. 1982. *The Syntax of Words*. Number 7 in Linguistic Inquiry Monograph Series. MIT Press.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *TACL*, 1:255–266.

Noah A. Smith and Mark Johnson. 2007. Weighted and probabilistic context-free grammars are equally expressive. *Computational Linguistics*, 33(4):477–491.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-level Chinese dependency parsing. In *ACL*, pages 1326–1336.

A Derivation of Eq. 6

Here we provide the gradient of the log-partition function as an expectation:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}(w) &= \frac{1}{Z_{\boldsymbol{\theta}}(w)} \nabla_{\boldsymbol{\theta}} Z_{\boldsymbol{\theta}}(w) \qquad (7) \\
&= \frac{1}{Z_{\boldsymbol{\theta}}(w)} \nabla_{\boldsymbol{\theta}} \left(\sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u',w)} \sum_{t' \in \mathcal{T}(u')} \mathit{score}_{\boldsymbol{\omega}}(t', u') \cdot \mathit{score}_{\boldsymbol{\eta}}(u', a, w) \right) \\
&= \frac{1}{Z_{\boldsymbol{\theta}}(w)} \sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u',w)} \sum_{t' \in \mathcal{T}(u')} \nabla_{\boldsymbol{\theta}} \left(\mathit{score}_{\boldsymbol{\omega}}(t', u') \cdot \mathit{score}_{\boldsymbol{\eta}}(u', a, w) \right) \\
&= \frac{1}{Z_{\boldsymbol{\theta}}(w)} \sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u',w)} \sum_{t' \in \mathcal{T}(u')} \Big(\mathit{score}_{\boldsymbol{\eta}}(u', a, w) \cdot \nabla_{\boldsymbol{\omega}} \mathit{score}_{\boldsymbol{\omega}}(t', u') \\
&\qquad + \mathit{score}_{\boldsymbol{\omega}}(t', u') \cdot \nabla_{\boldsymbol{\eta}} \mathit{score}_{\boldsymbol{\eta}}(u', a, w) \Big) \\
&= \frac{1}{Z_{\boldsymbol{\theta}}(w)} \sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u',w)} \sum_{t' \in \mathcal{T}(u')} \mathit{score}_{\boldsymbol{\eta}}(u', a, w) \cdot \mathit{score}_{\boldsymbol{\omega}}(t', u') \left(\sum_{\pi \in \Pi(t')} \boldsymbol{f}(\pi, u')^\top + \boldsymbol{g}(u', a, w)^\top \right) \\
&= \sum_{u' \in \Sigma^{|w|+k}} \sum_{a \in A(u',w)} \sum_{t' \in \mathcal{T}(u')} \frac{\mathit{score}_{\boldsymbol{\eta}}(u', a, w) \cdot \mathit{score}_{\boldsymbol{\omega}}(t', u')}{Z_{\boldsymbol{\theta}}(w)} \left(\sum_{\pi \in \Pi(t')} \boldsymbol{f}(\pi, u)^\top + \boldsymbol{g}(u', a, w)^\top \right) \\
&= \mathbb{E}_{(t,a,u) \sim p_{\boldsymbol{\theta}}(\cdot \mid w)} \left(\sum_{\pi \in \Pi(t)} \boldsymbol{f}(\pi, u)^\top + \boldsymbol{g}(u, a, w)^\top \right) \qquad (8)
\end{aligned}
$$

The result above can be further improved through Rao-Blackwellization. In this case, when we sample a tree–underlying form pair (t, u), we marginalize out all alignments that could have given rise to the sampled pair. The final derivation is show below:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}(w) &= \mathbb{E}_{(t,a,u) \sim p_{\boldsymbol{\theta}}(\cdot \mid w)} \left(\sum_{\pi \in \Pi(t)} \boldsymbol{f}(\pi, u)^\top + \boldsymbol{g}(u, a, w)^\top \right) \\
&= \mathbb{E}_{(t,u) \sim p_{\boldsymbol{\theta}}(\cdot \mid w)} \left(\sum_{\pi \in \Pi(t)} \boldsymbol{f}(\pi, u)^\top + \sum_{a \in A(u,w)} p_{\boldsymbol{\theta}}(a \mid u, w) \boldsymbol{g}(u, a, w)^\top \right) \qquad (9)
\end{aligned}
$$

This estimator in eq. (9) will have lower variance than eq. (8).