A Joint Model of Orthography and Morphological Segmentation

Ryan Cotterell **Tim Vieira**
Department of Computer Science
Johns Hopkins University, USA
{ryan.cotterell,tim.f.vieira}@gmail.com

Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
inquiries@cis.lmu.org

Abstract

We present a model of morphological segmentation that jointly learns to segment and restore orthographic changes, e.g., *funniest* ↦ *fun-y-est*. We term this form of analysis *canonical segmentation* and contrast it with the traditional *surface segmentation*, which segments a surface form into a sequence of substrings, e.g., *funniest* ↦ *funn-i-est*. We derive an importance sampling algorithm for approximate inference in the model and report experimental results on English, German and Indonesian.

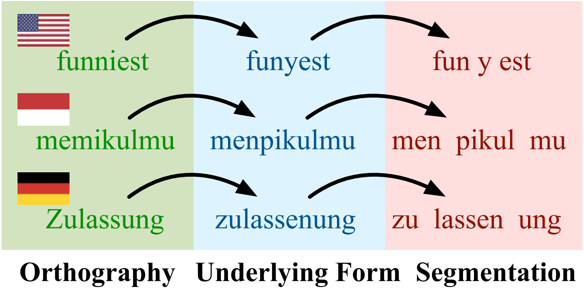

Figure 1: Examples of canonical segmentation for English (top), Indonesian (middle) and German (bottom).

1 Introduction

Morphological segmentation is useful for NLP applications, such as, 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). Prior work cast the problem as *surface segmentation:* a word form w is segmented into a sequence of substrings whose concatenation is w. In this paper, we introduce the problem of *canonical segmentation:* w is analyzed as a sequence of *canonical morphemes*, based on a set of word forms that have been "canonically" annotated for supervised learning. Each canonical morpheme c corresponds to a *surface morph* s, defined as its orthographic manifestation, i.e., as the substring of w that is generated by applying editing operations like insertion and deletion. Consider the following example: *funniest* has a canonical segmentation *fun-y-est* with three morphs *funn-i-est*. Arriving at the canonical analysis requires two edit operations: delete *n* in *funn* and replace *i* with *y* in *i*. Figure 1 gives examples of orthography (i.e., the concatentation of surface morphs), underlying form (i.e., the concatentation of canonical morphemes) and canonical segmentation in three languages.

Canonical segmentation is motivated in the following three ways: (i) Computational morphology is the study of how words and their meanings are composed from smaller units. This goal is better supported by canonical morphemes than by surface morphemes because the smaller units are more accurately modeled. For *funniest*, composition can reason with canonical morphemes *fun* and *y*, whereas surface segmentation must work with *funn* and *i*. (ii) Morphological analysis is typically done with attribute-value pairs (AVP), e.g., [`lemma`=FUNNY, `degree`=SUPER]. While AVP is a good representation for *inflectional* morphology, it is not powerful enough for *derivational* morphology. If we represent the derivation of *funnier* as [`lemma`=FUN, `deriv-suffix`=-Y, `degree`=SUPER], then it is no longer clear in this fixed representation whether

Proceedings of NAACL-HLT 2016, pages 664–669,
San Diego, California, June 12-17, 2016. ©2016 Association for Computational Linguistics

$\texttt{degree} = \text{SUPER}$ applies to *fun* or *fun*+*y*.[1] Canonical segmentation is more flexible—allowing us to express derivational relations without committing to a fixed attribute-value structure, which are used to study inflection. This point is important due to the fundamental distinction between the creation of words through inflection vs. through derivation. Inflection alters words to express syntactic relations (e.g., tense) with no major change in meaning nor POS. For example, *perturbed* and *perturbs* are inflections of the verb *perturb*. On the other hand, derivation modifies words more drastically—often changing the meaning or POS. For example, the noun *perturbation* derives from the verb stem *perturb* and the suffix *ation* (Haspelmath and Sims, 2013). (iii) Most NLP systems take *word forms* as atomic building blocks. We propose *canonical morphemes*, an alternative representation that models the structure of a language's lexicon and supports applications that benefit from access to the internal structure of words. This includes access to internal *morphological* structure, e.g., canonical morphemes like *-y* and *-ly* are recognized (independent of their orthographic manifestation) as derivational suffixes that cause predictable modifications; as well as access to internal *semantic* structure, e.g., the canonical segmentations of *fun* and *funny* share the canonical morpheme *fun*).

The contributions of this paper are as follows. We present the challenging new task of canonical segmentation. We develop a feature-rich structured joint model for canonical segmentation, which accounts for orthographic variation and segment-level structure. We derive an efficient importance sampling algorithm for approximate inference. We present experiments on three languages: English, German and Indonesian.

2 Model, Inference and Training

Our goal is canonical segmentation: identifying both the canonical morphemes and the morphs (their orthographic manifestations) of a word. This task involves *segmenting* the input as well as accounting for *orthographic* changes occurring in the word formation processes. Let w be the surface form, u the orthographic underlying representation (UR) of w, and s a labeled segmentation of u. Note: all random variables are string-valued (Dreyer and Eisner, 2009). For example, consider the word *unhappiness*:

$$\underbrace{\textit{unhappiness}}_{w} \overset{\text{transduction}}{\mapsto} \underbrace{\textit{unhappyness}}_{u}$$

$$\overset{\text{segmentation}}{\mapsto} \underbrace{[\text{ prefix } \textit{un}][\text{stem } \textit{happy}][\text{suffix } \textit{ness}]}_{s}.$$

Note that our notion of an orthographic UR closely resembles the phonological concept of a UR (Kenstowicz, 1994) and, indeed, many orthographic variations are manifestations of phonology.

We model this process as a globally normalized log-linear model of the conditional distribution,

$$p(s, u \mid w) = \frac{1}{Z(w)} \exp\Big(\boldsymbol{\eta}^\top \boldsymbol{f}(s, u) + \boldsymbol{\omega}^\top \boldsymbol{g}(u, w)\Big),$$

where $\boldsymbol{\theta} = \{\boldsymbol{\eta}, \boldsymbol{\omega}\}$ are the model parameters, $\boldsymbol{f}$ and $\boldsymbol{g}$ are, respectively, feature functions of the segmentation-UR and UR-surface-form pairs and $Z(w) = \sum_{s', u'} \exp\big(\boldsymbol{\eta}^\top \boldsymbol{f}(s', u') + \boldsymbol{\omega}^\top \boldsymbol{g}(u', w)\big)$ is the partition function. We can view this model as a conjunction of a finite-state transduction factor $\boldsymbol{g}$ (Dreyer et al., 2008) and a semi-Markov segmentation factor $\boldsymbol{f}$ (Sarawagi and Cohen, 2004), relating it to previous semi-CRF models of segmentation.[2] To fit the model, we maximize the log-likelihood of the training data $\{(s_i, u_i, w_i)\}_{i=1}^N$, $\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^N \log p(s_i, u_i \mid w_i)$, with respect to the model parameters $\boldsymbol{\theta}$. Optimization is done with gradient-based methods—requiring the computation of $\log Z(w)$ and $\nabla \log Z(w)$, which is intractable.[3] Thus, we turn to sampling (Rubinstein and Kroese, 2011) and stochastic gradient methods.

Features Our model includes several simple feature templates. The transduction factor of the model is based on (Cotterell et al., 2014): we include features that fire on individual edit actions as well as conjunctions of edit actions and characters on the surrounding context. For the semi-Markov factor, we use the feature set of Cotterell et al. (2015a), which

[1] Note that *funnest* is a word of (colloquial) English.

[2] Our transduction factor maps surface forms w to UR strings u of bounded length by imposing an insertion limit k. Thus, $|u| \leq |w| + k$. Our experiments use $k = 5$.

[3] Since the semi-CRF features fire on substrings, we would need a dynamic programming state for each substring of each of the exponentially many settings of u.

includes indicator features on individual segments, conjunctions of segments and segment labels and conjunctions of segments and left and right context on the input string. We also include a feature that checks whether the segment is a word in ASPELL (or a monolingual corpus).

Importance Sampling To approximately compute the gradient for learning, we employ importance sampling (MacKay, 2003, pp. 361–364). Rather than considering all underlying orthographic forms u, we use samples taken from proposal distribution q—a distribution over Σ^*. In the following equations, we omit the dependence on w for notational brevity. Also, let $\boldsymbol{h}(s,u) = \boldsymbol{f}(s,u) + \boldsymbol{g}(u,w)$. We now provide the derivation of our importance sampling estimate for the gradient of log-partition function, including Rao-Blackwellization (Robert and Casella, 2013).

$$\begin{aligned}
\nabla \log Z &= \mathop{\mathbb{E}}_{(s,u)\sim p} [\boldsymbol{h}(s,u)] && (1)\\
&= \sum_{s,u} p(s,u)\boldsymbol{h}(s,u) && (2)\\
&= \sum_{s,u} p(s|u)p(u)\boldsymbol{h}(s,u) && (3)\\
&= \sum_{u} p(u) \sum_{s} p(s|u)\boldsymbol{h}(s,u) && (4)\\
&= \sum_{u} p(u) \mathop{\mathbb{E}}_{s\sim p(\cdot|u)} [\boldsymbol{h}(s,u)] && (5)\\
&= \mathop{\mathbb{E}}_{u\sim q} \left[\frac{p(u)}{q(u)} \mathop{\mathbb{E}}_{s\sim p(\cdot|u)} [\boldsymbol{h}(s,u)] \right]. && (6)
\end{aligned}$$

The expectation $\mathbb{E}_{s\sim p(\cdot|u)} [\boldsymbol{h}(s,u)]$ is efficiently computed with the semi-Markov generalization of the forward-backward algorithm (Sarawagi and Cohen, 2004). The algorithm runs in $\mathcal{O}(n^2 \cdot t^2)$ per sample where n is the length of the string to be segmented and t is the size of the label space. In our case, we have three labels: prefix, stem and suffix so $t = 3$.

So long as q has support everywhere p does (i.e., $p(u) > 0 \Rightarrow q(u) > 0$), the estimate is unbiased. Unfortunately, we can only efficiently compute $p(u) \propto \sum_s \exp(\boldsymbol{\theta}^\top \boldsymbol{h}(s,u))$ up to constant factor, $p(u) = \bar{p}(u)/Z_u$. Thus, we use the *indirect importance sampling estimator*,

$$\frac{1}{\sum_{i=1}^{m} \frac{\bar{p}(u^{(i)})}{q(u^{(i)})}} \sum_{i=1}^{m} \frac{\bar{p}(u^{(i)})}{q(u^{(i)})} \mathop{\mathbb{E}}_{s\sim p(\cdot|u^{(i)})} \left[\boldsymbol{h}(s,u^{(i)}) \right], \quad (7)$$

where $u^{(1)} \ldots u^{(m)} \overset{\text{i.i.d.}}{\sim} q$. The indirect estimator is biased, but statistically consistent.[4] We also note that the particular instantiation of the indirect estimator leverages an efficient dynamic program to compute the expected features under $p(\cdot|u^{(i)})$. This has the effect of decreasing the number of samples required to get a useful estimate of the gradient. Computing $\bar{p}(u^{(i)})$ is a side effect of the dynamic program, namely the normalization constant. As a proposal distribution q, we use the following locally normalized distribution,

$$q(u) = \frac{\exp(\boldsymbol{\omega}^\top \boldsymbol{g}(u,w))}{\sum_{u'} \exp(\boldsymbol{\omega}^\top \boldsymbol{g}(u',w))}. \quad (8)$$

3 Related Work

Most work on morphological segmentation has been unsupervised. The LINGUISTICA (Goldsmith, 2001) and MORFESSOR (Creutz and Lagus, 2002) models rely on the minimum description length principle (Cover and Thomas, 2012). In short, these methods seek to segment words while at the same time minimizing the number of unique morphs discovered, i.e., the complexity of the model. The MORFESSOR model has additionally been augmented to handle the semi-supervised scenario (Kohonen et al., 2010). Goldwater et al. (2009) proposed a Bayesian non-parametric approach to word and morphological segmentation. Poon et al. (2009) used contrastive estimation (Smith and Eisner, 2005) to learn a log-linear model for segmentation fully unsupervised.

Few supervised techniques have been applied to morphological segmentation. Ruokolainen et al. (2013) applied a linear-chain CRF, showing that with a minimal amount of labeled data the performance of standard unsupervised and semi-supervised baselines are surpassed. In follow-up work (Ruokolainen et al., 2014), they found that incorporating distributional character-level features acquired from large unlabeled corpora improved the earlier model. Cotterell et al. (2015a) showed that modeling morphotactics with a semi-CRF improves results further.

The previously described approaches only attempt to split words into a sequence of stem and affixes—making it difficult to restore the underlying structure

[4] Informally, the indirect importance sampling estimate converges to the *true* expectation as $m \to \infty$.

which has been "corrupted" by the orthographic process. Our approach, however, is capable of restoring the underlying morphemes, e.g., *stopping* ↦ *stop-ing*. We note two exceptions to the above statement. Both Dasgupta and Ng (2007) and Naradowsky and Goldwater (2009) incorporate basic, heuristic spelling rules into *unsupervised* induction algorithms. Relatedly, Cotterell et al. (2015b) induced a phonology in an unsupervised manner. In contrast, our model is fully supervised and supports rich features, which enable accurate prediction on new words.

4 Experiments

We provide canonical segmentation experiments in three languages: English, German and Indonesian.

4.1 Corpora

The English data was extracted from segmentations derived from CELEX (Baayen et al., 1993). The German data was extracted from DerivBase (Zeller et al., 2013), which provides a collection of derived forms and the transformation rules. We manipulated these rules to create canonical segmentations. Lastly, the Indonesian data was created from the output of the MORPHIND analyzer (Larasati et al., 2011), which we ran on an open-source corpus of Indonesian.[5] For each language we selected 10,000 forms at random from a uniform distribution over types to form our corpus. We sampled 5 splits of the data into 8000 training forms, 1000 development forms and 1000 test forms. We have released all train, development and test splits online with additional documentation about their construction.[6]

4.2 Models

We train two versions of our proposed model. First, we train a *pipeline model*, i.e., we train the transduction component and segmentation component independently and decode sequentially. This approach is faster both at train and at test but suffers from cascading errors. Second, we train a *joint model*, the transduction and the segmentation components are trained to work well together.

		Joint	Pipeline	SemiCRF	WFST
error	en	**0.27** (.02)	0.33 (.01)	0.33 (.01)	0.63 (.00)
	de	**0.41** (.03)	0.53 (.02)	0.65 (.01)	0.74 (.01)
	id	**0.10** (.01)	0.22 (.01)	0.27 (.01)	0.71 (.00)
distance	en	0.98 (.34)	**0.63** (.04)	0.68 (.01)	1.35 (.01)
	de	**1.01** (.07)	1.10 (.04)	1.32 (.04)	4.24 (.20)
	id	**0.15** (.02)	0.36 (.03)	0.49 (.02)	2.13 (.00)
F_1	en	**0.76** (.02)	0.70 (.02)	0.68 (.01)	0.53 (.02)
	de	**0.76** (.02)	0.71 (.01)	0.65 (.01)	0.59 (.02)
	id	**0.80** (.01)	0.75 (.01)	0.71 (.01)	0.62 (.02)

Table 1: Top: Error rate. Middle: Average edit distance. Bottom: Mean morpheme F_1 (higher better). Standard deviation in parentheses. Best result on each line in bold.

Baseline: Semi-CRF Segmenter The first baseline is a semi-CRF (Sarawagi and Cohen, 2004) that segments the orthographic form into morphs *without* canonicalization. Earlier work by Cotterell et al. (2015a) applied this model to *supervised* morphological segmentation. We use the feature set as Cotterell et al. (2015a), but we do not incorporate their augmented morphotactic state space.

Baseline: WFST Segmenter Our second baseline is a weighted finite-state transducer (Mohri, 1997) with a log-linear parameterization (Dreyer et al., 2008). We use the stochastic contextual edit model of Cotterell et al. (2014). We employ context n-gram features (up to 6-grams) on the input string to the left and right of the edit location in addition to 2-gram features on the lower string. The context features are then conjoined with the exact edit action. We refer the reader to Cotterell et al. (2014) for more details. The segmentation boundaries are marked as a distinguished symbol in the target string. This model is not entirely suited for the task as it makes it difficult to include the rich features we get through ASPELL.

Training and Decoding Details We train all models with AdaGrad (Duchi et al., 2011; Bottou, 2010). For the joint model, we take 10 samples ($m = 10$) for each gradient estimate. See Algorithm 3 of Bengio et al. (2003) for pseudocode for SGD with importance sampling. The pipeline and segmentation models use ordinary SGD. We use L_2 regularization with the regularization coefficient chosen by based on development set performance.

Exact decoding, $\operatorname{argmax}_{s,u} p(s, u \mid w)$, is intractable. Thus, we use a sampling approximation:

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

[6] http://ryancotterell.github.io/canonical-segmentation/

$\text{argmax}_{s,u^{(i)}}\, p(s, u^{(i)} \mid w)$ where $u^{(1)} \ldots u^{(m)} \overset{\text{i.i.d.}}{\sim} q$. We use $m = 1000$ in our experiments. Conditioned on each sample value for u, we use exact semi-CRF Viterbi decoding to select s.

4.3 Evaluation Measures

Evaluating morphological segmentation is tricky. The standard measure for the supervised task is border F_1, which measures how often the segmentation boundaries posited by the model are correct. However, this measure assumes that the concatenation of the segments is identical to the input string (i.e., surface segmentation) and is thus not applicable to canonical segmentation. On the other hand, the Morpho Challenge competition (Kurimo et al., 2010) uses a measure that samples a large number of word pairs from a linguistic gold standard. A form is considered correct if the gold standard contains at least one overlapping morph *and* the model posits at least one overlapping morph—this is problematic because for languages with multi-morphemic words (e.g., German), one should consider all morphs. Moreover, we can actually recover the linguistically annotated gold standard in contrast to unsupervised methods.

Instead, we report results under three measures: error rate, edit distance and morpheme F_1. Error rate is the proportion of analyses that are completely correct. Since error rate gives no partial credit, we also report edit distance between the predicted analysis and the gold standard, where both are encoded as strings using a distinguished boundary character at segment boundaries. Finally, morpheme F_1 (van den Bosch and Daelemans, 1999) considers overlap between the *set* of morphemes in the model's analysis and the set of morphemes in the gold standard. In this case, precision asks how often did the predicted segmentation contain morphemes in the gold standard and recall asks how often were the gold standard morphemes in the predicted segmentation.

4.4 Results and Error Analysis

Table 1 gives results for the three measures. Under error rate and morpheme F_1 our joint model performs the best on all three languages, followed by our pipeline model and then the two baselines. In fact, we observe that error rate and F_1 are quite correlated in general. Under edit distance, the joint model is the best model on German and Indonesian, but the pipeline model is superior on English. Error analysis indicates that the lower performance is due to spurious insertions. For example, our model incorrectly analyzes *ruby* (stone) as *ruble-y*, mistaking the *ruby* as an adjectival form of *ruble* (the Russian currency); the correct analysis is *ruby* $\mapsto$ *ruby*. We believe that a richer transduction component may fix some of these problems. Overall, our joint model performs well; it is on average within one edit operation of the gold segmentation on three languages.

Unsurprisingly, the WFST performs poorly because it cannot leverage segment-level features (e.g., ASPELL features), which are available to the other models. The performance of the semi-CRF is limited by the orthographic changes in the language, which it cannot model. German is rich in such changes, hence the semi-CRF performs poorly and gets more than half the test cases wrong.

5 Conclusion

We presented a joint model for the task of canonical morphological segmentation, which extends existing approaches with the ability to learn orthographic changes. We argue that canonical morphological segmentation provides a useful analysis of linguistic phenomena (e.g., derivational morphology) because the sequence of morphemes is canonical—making it evident, which words share morphemes. Our model outperforms two baselines on three languages.

Acknowledgments

This material is based in part on research sponsored by DARPA under agreement number FA8750-13-2-0017 (the DEFT program) and the National Science Foundation under Grant No. 1423276. RC was funded by a DAAD Long-Term Research Grant. HS was supported by DFG (SCHU 2246/4-2).

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*.

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

Yoshua Bengio, Jean-Sébastien Senécal, et al. 2003.

Quick training of probabilistic neural nets by importance sampling. In *AISTATS*.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

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, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015a. Labeled morphological segmentation with semi-Markov models. In *CoNLL*.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015b. Modeling word forms using latent underlying morphs and phonology. *TACL*, 3:433–447.

Thomas M Cover and Joy A Thomas. 2012. *Elements of information theory*. John Wiley & Sons.

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

Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *NAACL*.

Markus Dreyer and Jason Eisner. 2009. Graphical models over multiple strings. In *EMNLP*.

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. *The Journal of Machine Learning Research*, 12:2121–2159.

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

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Martin Haspelmath and Andrea Sims. 2013. *Understanding morphology*. Routledge.

Michael Kenstowicz. 1994. *Phonology in Generative Grammar*. Blackwell.

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

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho Challenge competition 2005–2010: evaluations and results. In *SIGMORPHON*.

Septina Dian Larasati, Vladislav Kuboň, and Daniel Zeman. 2011. Indonesian morphology tool (morphind): Towards an Indonesian corpus. In *Systems and Frameworks for Computational Morphology*, pages 119–129. Springer.

David JC MacKay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311.

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*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL*. Association for Computational Linguistics.

Christian Robert and George Casella. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.

Reuven Y Rubinstein and Dirk P Kroese. 2011. *Simulation and the Monte Carlo method*. John Wiley & Sons.

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

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

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

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

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*.

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

Britta D Zeller, Jan Snajder, and Sebastian Padó. 2013. DErivBase: Inducing and evaluating a derivational morphology resource for german. In *ACL*.