A Comparative Study of Minimally Supervised Morphological Segmentation

Teemu Ruokolainen*
Aalto University

Oskar Kohonen**
Aalto University

Kairit Sirts†
Tallinn University of Technology

Stig-Arne Grönroos*
Aalto University

Mikko Kurimo*
Aalto University

Sami Virpioja**
Aalto University

This article presents a comparative study of a subfield of morphology learning referred to as **minimally supervised morphological segmentation**. *In morphological segmentation, word forms are segmented into morphs, the surface forms of morphemes. In the minimally supervised data-driven learning setting, segmentation models are learned from a small number of manually annotated word forms and a large set of unannotated word forms. In addition to providing a literature survey on published methods, we present an in-depth empirical comparison on three diverse model families, including a detailed error analysis. Based on the literature survey, we conclude that the existing methodology contains substantial work on generative morph lexicon-based approaches and methods based on discriminative boundary detection. As for which approach has been more successful, both the previous work and the empirical evaluation presented here strongly imply that the current state of the art is yielded by the discriminative boundary detection methodology.*

* Department of Signal Processing and Acoustics, Otakaari 5 A, FI-02150 Espoo, Finland.
E-mail: {teemu.ruokolainen, stig-arne.gronroos, mikko.kurimo}@aalto.fi.
** Department of Information and Computer Science, Konemiehentie 2, FI-02150 Espoo, Finland.
E-mail: {oskar.kohonen, sami.virpioja}@aalto.fi.
† Institute of Cybernetics, Akadeemia tee 21 EE-12618 Tallin, Estonia. E-mail: sirts@phon.ioc.ee.

Submission received: 15 September 2014; revised version received: 7 October 2015; accepted for publication: 15 November 2015

doi:10.1162/COLI_a_00243

1. Introduction

This article discusses a subfield of morphology learning referred to as **morphological segmentation**, in which word forms are segmented into **morphs**, the surface forms of **morphemes**. For example, consider the English word *houses* with a corresponding segmentation *house+s*, where the segment *house* corresponds to the word stem and the suffix *-s* marks the plural number. Although this is a major simplification of the diverse morphological phenomena present in languages, this type of analysis has nevertheless been of substantial interest to computational linguistics, beginning with the pioneering work on morphological learning by Harris (1955). As for automatic language processing, such segmentations have been found useful in a wide range of applications, including speech recognition (Hirsimäki et al. 2006; Narasimhan et al. 2014), information retrieval (Turunen and Kurimo 2011), machine translation (de Gispert et al. 2009; Green and DeNero 2012), and word representation learning (Luong, Socher, and Manning 2013; Qiu et al. 2014).

Since the early work of Harris (1955), most research on morphological segmentation has focused on **unsupervised** learning, which aims to learn the segmentation from a list of unannotated (unlabeled) word forms. The unsupervised methods are appealing as they can be applied to any language for which there exists a sufficiently large set of unannotated words in electronic form. Consequently, such methods provide an inexpensive means of acquiring a type of morphological analysis for low-resource languages as motivated, for example, by Creutz and Lagus (2002). The unsupervised approach and learning setting has received further popularity because of its close relationship with the unsupervised **word segmentation** problem, which has been viewed as a realistic setting for theoretical study of language acquisition (Brent 1999; Goldwater 2006).

Although development of novel unsupervised model formulations has remained a topic of active research (Poon, Cherry, and Toutanova 2009; Monson, Hollingshead, and Roark 2010; Spiegler and Flach 2010; Lee, Haghighi, and Barzilay 2011; Sirts and Goldwater 2013), recent work has also shown a growing interest towards **semi-supervised** learning (Poon, Cherry, and Toutanova 2009; Kohonen, Virpioja, and Lagus 2010; Sirts and Goldwater 2013; Grönroos et al. 2014; Ruokolainen et al. 2014). In general, the aim of semi-supervised learning is to acquire high-performing models utilizing both unannotated as well as annotated data (Zhu and Goldberg 2009). In morphological segmentation, the annotated data sets are commonly small, on the order of a few hundred word forms. We refer to this learning setting with such a small amount of supervision as **minimally supervised** learning. In consequence, similar to the unsupervised methods, the minimally supervised techniques can be seen as a means of acquiring a type of morphological analysis for under-resourced languages.

Individual articles describing novel methods typically contain a comparative discussion and empirical evaluation between one or two preceding approaches. Therefore, what is currently lacking from the literature is a summarizing comparative study on the published methodology as a whole. Moreover, the literature currently lacks discussion on error analysis. A study on the error patterns produced by varying approaches could inform us about their potential utility in different tasks. For example, if an application requires high-accuracy compound splitting, one could choose to apply a model with a good compound-splitting capability even if its affix accuracy does not reach state of the art. The purpose of this work is to address these issues.

Our main contributions are as follows. First, we present a literature survey on morphological segmentation methods applicable in the minimally supervised learning setting. The considered methods include unsupervised techniques that learn solely from

unannotated data, supervised methods that utilize solely annotated data, and semi-supervised approaches that utilize both unannotated and annotated data. Second, we perform an extensive empirical evaluation of three diverse method families, including a detailed error analysis. The approaches considered in this comparison are variants of the Morfessor algorithm (Creutz and Lagus 2002, 2005, 2007; Kohonen, Virpioja, and Lagus 2010; Grönroos et al. 2014), the adaptor grammar framework (Sirts and Goldwater 2013), and the conditional random field method (Ruokolainen et al. 2013, 2014). We hope the presented discussion and empirical evaluation will be of help for future research on the considered task.

The rest of the article is organized as follows. In Section 2, we provide an overview of related studies. We then provide a literature survey of published morphological segmentation methodology in Section 3. Experimental work is presented in Section 4. Finally, we provide a discussion on potential directions for future work and conclusions on the current work in Sections 5 and 6, respectively.

2. Related Work

Hammarström and Borin (2011) presented a literature survey on unsupervised learning of morphology, including methods for learning morphological segmentation. Whereas the discussion provided by Hammarström and Borin focuses mainly on linguistic aspects of morphology learning, our work is strongly rooted in machine learning methodology and empirical evaluation. In addition, whereas Hammarström and Borin focus entirely on unsupervised learning, our work considers a broader range of learning paradigms. Therefore, although related, Hammarström and Borin and our current presentation are complementary in that they have different focus areas.

In addition to the work of Hammarström and Borin (2011), we note that there exists some established forums on morphology learning. First, we mention the ACL Special Interest Group on Computational Morphology and Phonology (SIGMORPHON), which has regularly organized workshops on the subject since 2002. As for specifically morphology learning, we refer to the Morpho Challenge competitions organized since 2005 at Aalto University (formerly known as Helsinki University of Technology). Although these events have been successful in providing a publication and discussion venue for researchers interested in the topic, they have not given birth to comparative studies or survey literature. For example, whereas the publications on the Morpho Challenge (Kurimo et al. 2009; Kurimo, Virpioja, and Turunen 2010) discuss the competition results, they nevertheless do not attempt to provide any insight on the fundamental differences and similarities of the participating methods.

3. Methods

This section provides a detailed review of our methodology. We begin by describing varying morphological representations, including segmentation, and the minimally supervised learning setting in Sections 3.1 and 3.2, respectively. We then provide a literature survey and comparative discussion on a range of methods in Section 3.3.

3.1 On Learning Morphological Representations

In what follows, we briefly characterize morphological segmentation with respect to alternative morphological representations, particularly the **full morphological analysis**. To this end, consider the exemplar segmentations and full analyses for Finnish

Table 1
Morphological segmentation versus full morphological analysis for exemplar Finnish word forms. The full analysis consists of word lemma (basic form), part-of-speech, and fine-grained labels.

word form	full analysis	segmentation
auto (car)	auto+N+Sg+Nom	auto
autossa (in car)	auto+N+Sg+Ine	auto+ssa
autoilta (from cars)	auto+N+Pl+Abl	auto+i+lta
autoilta (car evening)	auto+N+Sg+Nom+# ilta+N+Sg+Nom	auto+ilta
maantie (highway)	maantie+N+Sg+Nom	maantie
	maa+N+Sg+Gen+# tie+N+Sg+Nom	maa+n+tie
sähköauto (electric car)	sähköauto+N+Sg+Nom	sähköauto
	sähkö+N+Sg+Nom+# auto+N+Sg+Nom	sähkö+auto

word forms in Table 1, where the full analyses are provided by the rule-based OMorFi analyzer developed by Pirinen (2008). Note that it is typical for word forms to have alternative analyses and/or meanings that cannot be disambiguated without sentential context. Evidently, the level of detail in the full analysis is substantially higher compared with the segmentation, as it contains lemmatization as well as morphological tagging, whereas the segmentation consists of only segment boundary positions. Consequently, because of this simplification, morphological segmentation has been amenable to unsupervised machine learning methodology, beginning with the work of Harris (1955). Meanwhile, the majority of work on learning of full morphological analysis has used supervised methodology (Chrupala, Dinu, and van Genabith 2008). Lastly, there have been numerous studies on statistical learning of intermediate forms of segmentation and full analysis (Lignos 2010; Virpioja, Kohonen, and Lagus 2010) as well as alternative morphological representations (Yarowsky and Wicentowski 2000; Schone and Jurafsky 2001; Neuvel and Fulop 2002; Johnson and Martin 2003).

As for language processing, learning segmentation can be advantageous compared with learning full analyses. In particular, learning full analysis in a supervised manner typically requires up to tens of thousands of manually annotated sentences. A low-cost alternative, therefore, could be to learn morphological segmentation from unannotated word lists and a handful of annotated examples. Importantly, segmentation analysis has been found useful in a range of applications, such as speech recognition (Hirsimäki et al. 2006; Narasimhan et al. 2014), information retrieval (Turunen and Kurimo 2011), machine translation (de Gispert et al. 2009; Green and DeNero 2012), and word representation learning (Luong, Socher, and Manning 2013; Qiu et al. 2014).

Despite its intuitiveness, it should be noted that the segmentation representation is not equally applicable to all languages. To this end, consider the terms **isolative** and **synthetic** languages. In languages with a high amount of isolating morphological properties, word forms tend to comprise their own morphemes. Meanwhile, in heavily synthetic languages, words tend to contain multiple morphemes. Synthetic languages can be described further according to their **agglutinative** (**concatenative**) and **fusional** properties. In the former, the morphs tend to have clear boundaries between them whereas in the latter, the morphs tend to be indistinguishable. For examples of agglutinative and fusional word formation, consider the English verbs *played* (past tense of *play*) and *sang* (past tense of *sing*). Where the previous can be effortlessly

divided into two segments as *play+ed* (STEM + PAST TENSE), there are no such distinct boundaries in the latter. Generally, languages with synthetic properties mix concatenative and fusional schemes and contain agglutinative properties to varying degrees. Morphological segmentation can be most naturally applied to highly agglutinative languages.

Morphologically ambiguous word forms are common especially in highly synthetic languages. Even without disambiguation based on sentential context, providing all correct alternatives could be useful for some downstream applications, such as information retrieval. Statistical methods can usually provide n-best segmentations; for example, Morfessor (Creutz and Lagus 2007) and CRFs (Ruokolainen et al. 2013) by using n-best Viterbi algorithm and adaptor grammar (Sirts and Goldwater 2013) by collecting the variations in the posterior distribution samples. Although there is no evident way to decide the correct number of alternatives for a particular word form, n-best lists might be useful whenever recall (including the correct answers) is more important than precision (excluding any incorrect answers). The Morpho Challenge competitions have allowed providing alternative segmentations for the submitted methods, but no clear developments have been reported. In fact, even in the reference results based on the gold standard segmentations, selecting all alternative segmentations has performed slightly worse in the information retrieval tasks than taking only the first segmentation (Kurimo, Virpioja, and Turunen 2010).

3.2 Minimally Supervised Learning Settings

In data-driven morphological segmentation, our aim is to learn segmentation models from training data. Subsequent to training, the models provide segmentations for given word forms. In the minimally supervised learning setting, as defined here, the models are estimated from annotated and unannotated word forms. We denote the annotated data set comprising word forms with their corresponding segmentation as $\mathcal{D}$ and the unannotated data set comprising raw word forms as $\mathcal{U}$. Typically, the raw word forms can be obtained easily and, consequently, $\mathcal{U}$ can contain millions of word forms. Meanwhile, acquiring the annotated data $\mathcal{D}$ requires manual labor and, therefore, typically contains merely hundreds or thousands of word forms. For an illustration of $\mathcal{D}$ and $\mathcal{U}$, see Table 2.

Table 2
Examples of annotated and unannotated data, $\mathcal{D}$ and $\mathcal{U}$, respectively. Typically, $\mathcal{U}$ can contain hundreds of thousands or millions of word forms, whereas $\mathcal{D}$ contains merely hundreds or thousands of word forms.

$\mathcal{D}$	$\mathcal{U}$
anarch + ist + s	actions
bound + ed	bilinguals
conting + ency	community
de + fame	disorders
entitle + ment	equipped
fresh + man	faster
...	...

We consider three machine learning approaches applicable in the minimally supervised learning setting, namely, unsupervised, supervised, and semi-supervised learning. In unsupervised learning, the segmentation models are trained on solely unannotated data $\mathcal{U}$. Meanwhile, supervised models are trained from solely the annotated data $\mathcal{D}$. Finally, the aim of semi-supervised learning is to utilize both the available unannotated and annotated data. Because the semi-supervised approach utilizes the largest amount of data, it is expected to be most suitable for acquiring high segmentation accuracy in the minimally supervised learning setting.

Lastly, we note that the unsupervised learning framework can be understood in a strict or non-strict sense, depending on whether the applied methods are allowed to use annotated data $\mathcal{D}$ for hyperparameter tuning. Although the term unsupervised learning itself suggests that such adjusting is infeasible, this type of tuning is nevertheless common (Creutz et al. 2007; Çöltekin 2010; Spiegler and Flach 2010; Sirts and Goldwater 2013). In addition, the minimally supervised learning setting explicitly assumes a small amount of available annotated word forms. Consequently, in the remainder of this article, all discussion on unsupervised methods refers to unsupervised learning in the non-strict sense.

3.3 Algorithms

Here we provide a literature survey on proposed morphological segmentation methods applicable in the minimally supervised learning setting. We place particular emphasis on three method families, namely, the Morfessor algorithm (Creutz and Lagus 2002, 2005, 2007; Kohonen, Virpioja, and Lagus 2010; Grönroos et al. 2014), the adaptor grammar framework (Sirts and Goldwater 2013), and conditional random fields (Ruokolainen et al. 2013, 2014). These approaches are the subject of the empirical evaluation presented in Section 4. We present individual method descriptions in Section 3.3.1. Subsequently, Section 3.3.2 provides a summarizing discussion, the purpose of which is to gain insight on the fundamental differences and similarities between the varying approaches.

3.3.1 Descriptions

Morfessor. We begin by describing the original, unsupervised Morfessor method family (Creutz and Lagus 2002, 2005, 2007). We then discuss the later, semi-supervised extensions (Kohonen, Virpioja, and Lagus 2010; Grönroos et al. 2014). In particular, we review the extension of Morfessor Baseline to semi-supervised learning by using a weighted generative model (Kohonen, Virpioja, and Lagus 2010), and then discuss the most recent Morfessor variant, FlatCat (Grönroos et al. 2014). Finally, we discuss some general results from the literature on semi-supervised learning with generative models.

The unsupervised Morfessor methods are based on a generative probabilistic model that generates the observed word forms $x_i \in \mathcal{U}$ by concatenating morphs $x_i = m_{i1} \circ m_{i2} \circ \cdots \circ m_{in}$. The morphs are stored in a **morph lexicon**, which defines the probability of each morph $P(m \mid \theta)$ given some parameters θ. The Morfessor learning problem is to find a morph lexicon that strikes an optimal balance between encoding the observed word forms concisely and, at the same time, having a concise morph lexicon. To this end, Morfessor utilizes a prior distribution $P(\theta)$ over morph lexicons, derived from the Minimum Description Length principle (Rissanen 1989), that favors lexicons that contain fewer, shorter morphs. This leads to the following minimization problem

that seeks to balance the conciseness of the lexicon with the conciseness of the observed corpus encoded with the lexicon:

$$\theta^* = \arg\min_{\theta} L(\theta, \mathcal{U}) = \arg\min_{\theta} \{-\ln P(\theta) - \ln P(\mathcal{U} \mid \theta)\}, \quad (1)$$

The optimization problem in Equation (1) is complicated by the fact that each word in the corpus $\mathcal{U}$ can be generated by different combinations of morphs, defined by the set of segmentations of that word. This introduces a nuisance parameter z for the segmentation of each word form, where $P(\mathcal{U} \mid \theta) = \sum_z P(\mathcal{U} \mid z, \theta) P(z)$. Because of this summation, the expression cannot be solved analytically, and iterative optimization must be used instead.

The unsupervised Morfessor variants differ in the following ways: first, whether all morphs belong to a single category or the categories PREFIX, STEM, and SUFFIX are used; secondly, if the model utilizes a lexicon that is **flat** or **hierarchical**. In a flat lexicon, morphs can only be encoded by combining letters, whereas in a hierarchical lexicon pre-existing morphs can be used for storing longer morphs. Thirdly, the parameter estimation and inference methods differ. Parameters are estimated using greedy local search or iterative batch procedures while inference is performed with either Viterbi decoding or heuristic procedures.

The earliest Morfessor method, referred to as Morfessor Baseline, has been extended to semi-supervised learning by Kohonen, Virpioja, and Lagus (2010). In contrast, the later methods, namely, Categories-ML and Categories-MAP, have not been extended, as they use either hierarchical lexicons or training procedures that make them less amenable to semi-supervised learning. However, recently Grönroos et al. (2014) proposed a new Morfessor variant that uses morph categories in combination with a flat lexicon, and can therefore apply the semi-supervised learning technique of Kohonen, Virpioja, and Lagus.

We begin the description of the semi-supervised extension to Morfessor Baseline (Creutz and Lagus 2002, 2007) by reviewing its generative model. Morfessor Baseline utilizes a model in which word forms are generated by concatenating morphs, all of which belong to the same category. It utilizes a flat morph lexicon $P(m \mid \theta)$ that is simply a multinomial distribution over morphs m, according to the probabilities given by the parameter vector θ. The utilized prior penalizes storing long morphs in the lexicon by assigning each stored morph a cost that depends most strongly on the morph length in letters. A morph is considered to be stored if the lexicon assigns it a nonzero probability. The parameter estimation for θ finds a local optimum utilizing greedy local search. The search procedure approximates the optimization problem in Equation (1) by assuming that, for each word form x_i, its corresponding segmentation distribution $P(z_i)$ has all its mass concentrated to a single segmentation z_i. The parameter estimation is then performed by locally searching each word for the segmentation that yields the best value of the cost function in Equation (1). The process is repeated for all words in random order until convergence. Subsequent to learning, the method predicts the segmentation of a word form by selecting the segmentation with the most probable sequence of morphs using an extension of the Viterbi algorithm.

Semi-supervised learning is in principle trivial for a generative model: For the labeled word forms $\mathcal{D}$, the segmentation is fixed to its correct value, and for the unlabeled forms $\mathcal{U}$ the standard parameter estimation procedure is applied. However, Kohonen,

Virpioja, and Lagus (2010) failed to achieve notable improvements in this fashion, and consequently replaced the minimized function L in Equation (1) with

$$L(\theta, z, \mathcal{U}, \mathcal{D}) = -\ln P(\theta) - \alpha \times \ln P(\mathcal{U} \mid \theta) - \beta \times \ln P(\mathcal{D} \mid \theta). \quad (2)$$

Such weighted objectives were used earlier in combination with generative models by, for example, Nigam et al. (2000). The semi-supervised training procedure then adjusts the weight values α and β. The absolute values of the weights control the cost of encoding a morph in the training data with respect to the cost of adding a new morph to the lexicon, and their ratio controls how much weight is placed on the annotated data with respect to the unannotated data. When the hyperparameters α and β are fixed, the lexicon parameters θ can be optimized with the same greedy local search procedure as in the unsupervised Morfessor Baseline. The weights can then be optimized with a grid search and by choosing the model with the best evaluation score on a held-out development set. Although this modification is difficult to justify from the perspective of generative modeling, Kohonen, Virpioja, and Lagus show that in practice it can yield performance improvements. From a theoretical point of view, it can be seen as incorporating discriminative training techniques when working with a generative model by optimizing for segmentation performance rather than maximum a posteriori probability. However, only the hyperparameters are optimized in this fashion, whereas the lexicon parameters are still learned within the generative model framework.

The semi-supervised learning strategy described here is simple to apply if the objective function in Equation (1) can be factored to parts that encode the morphs using letters and encode the training corpus using the morphs. For some models of the Morfessor family this is not possible because of the use of a hierarchical lexicon, where morphs can be generated from other morphs as well as from individual letters. In particular, this includes the well-performing Categories-MAP variant (Creutz et al. 2007). In contrast to Morfessor Baseline, the Categories-MAP and the preceding Categories-ML method use a hidden Markov model to produce the observed words, where the states are given by STEM, PREFIX, SUFFIX categories as well as an internal non-morpheme category. A recent development is Morfessor FlatCat by Grönroos et al. (2014), which uses the hidden Markov model structure and morph categories in combination with a flat lexicon, thus allowing semi-supervised learning in the same fashion as for Morfessor Baseline.

In general, the key idea behind using the weighted objective function in Equation (2) for semi-supervised learning is that the hyperparameters α and β can be used to explicitly control the influence of the unannotated data on the learning. Similar semi-supervised learning strategies have also been used in other problems. For classification with generative models, it is known that adding unlabeled data to a model trained with labeled data can degrade performance (Cozman et al. 2003; Cozman and Cohen 2006). In particular, this can be the case if the generative model does not match the generating process, something that is difficult to ensure in practice. Recently, this phenomenon was analyzed in more detail by Fox-Roberts and Rosten (2014), who show that, although the unlabeled data can introduce a bias, the bias can be removed by optimizing a weighted likelihood function where the unlabeled data is raised to the power $\frac{N_L}{N}$, where N_L is the number of labeled samples and N is the number of all samples. This corresponds to the weighting scheme used in Morfessor when setting the ratio $\frac{\alpha}{\beta} = \frac{N_L}{N}$.

Adaptor Grammars. Recently, Sirts and Goldwater (2013) presented work on minimally supervised morphological segmentation using the adaptor grammar (AG) approach (Johnson, Griffiths, and Goldwater 2006). The AGs are a non-parametric Bayesian modeling framework applicable for learning latent tree structures over an input corpus of strings. They can be used to define morphological grammars of different complexity, starting from the simplest grammar where each word is just a sequence of morphs and extending to more complex grammars, where each word consists, for example, of zero or more prefixes, a stem, and zero or more suffixes.

The actual forms of the morphs are learned from the data and, subsequent to learning, used to generate segmentations for new word forms. In this general approach, AGs are similar to the Morfessor family (Creutz and Lagus 2007). A major difference, however, is that the morphological grammar is not hard-coded but instead specified as an input to the algorithm. This allows different grammars to be explored in a flexible manner. Prior to the work by Sirts and Goldwater, the AGs were successfully applied in a related task of segmenting utterances into words (Johnson 2008; Johnson and Goldwater 2009; Johnson and Demuth 2010).

The second major difference between the Morfessor family and the AG framework is the contrast between the MAP and fully Bayesian estimation approaches. Whereas the search procedure of the Morfessor method discussed earlier returns a single model corresponding to the MAP point-estimate, AGs instead operate with full posterior distributions over all possible models. Because acquiring the posteriors analytically is intractable, **inference** is performed utilizing Markov chain Monte Carlo algorithms to obtain samples from the posterior distributions of interest (Johnson 2008; Johnson and Goldwater 2009; Johnson and Demuth 2010; Sirts and Goldwater 2013). However, as sampling-based models are costly to train on large amounts of data, we adopt the parsing-based method proposed in Sirts and Goldwater (2013) to use the trained AG model inductively on test data. One of the byproducts of training the AG model is the **posterior grammar**, which in addition to all the initial grammar rules, also contains the cached subtrees learned by the system. This grammar can be used in any standard parser to obtain segmentations for new data.

The AG framework was originally designed for the unsupervised learning setting, but Sirts and Goldwater (2013) introduced two approaches for semi-supervised learning they call the semi-supervised AG and AG Select methods. The semi-supervised AG approach is an extension to unsupervised AG, in which the annotated data $\mathcal{D}$ is exploited in a straightforward manner by keeping the annotated parts of parse trees fixed while inferring latent structures for the unannotated parts. For unannotated word forms, inference is performed on full trees. For example, the grammar may specify that words are sequences of morphs and each morph is a sequence of submorphs. Typically, the annotated data only contain morpheme boundaries and submorphs are latent in this context. In this situation the inference for annotated data is performed over submorph structures only.

Similarly to unsupervised learning, semi-supervised AG requires the morphological grammar to be defined manually. Meanwhile, the AG Select approach aims to automate the grammar development process by systematically evaluating a range of grammars and finding the best one. AG Select is trained using unsupervised AG with an uninformative **metagrammar** so that the resulting parse-trees contain many possible segmentation templates. To find out which template works the best for any given language or data set, each of these templates are evaluated using the annotated data set $\mathcal{D}$. In this sense, AG Select can be characterized as more of a model selection method than semi-supervised learning.

Conditional Random Fields. The Morfessor and AG algorithms discussed earlier, although different in several respects, operate in a similar manner in that they both learn lexicons. For Morfessor, the lexicon consists of morphs, whereas for AG, the lexical units are partial parse-trees. Subsequent to learning, new word forms are segmented either by generating the most likely morph sequences (Morfessor) or by sampling parse trees from the posterior distribution (AG). In what follows, we consider a different approach to segmentation using sequence labeling methodology. The key idea in this approach is to focus the modeling effort to **morph boundaries** instead of the whole segments. Following the presentation of Ruokolainen et al. (2013, 2014), the morphological segmentation task can be represented as a sequence labeling problem by assigning each character in a word form to one of three classes, namely,

B	beginning of a multi-character morph
M	middle of a multi-character morph
S	single-character morph

Using this label set, one can represent the segmentation of the Finnish word *autoilta* (*from cars*) (*auto+i+lta*) as

a	u	t	o	i	l	t	a
↓	↓	↓	↓	↓	↓	↓	↓
B	M	M	M	S	B	M	M

Naturally, one can also use other label sets. Essentially, by defining more fine-grained labels, one captures increasingly eloquent structure but begins to overfit model to the training data because of increasingly sparser statistics. Subsequent to defining the label set, one can learn a segmentation model using general sequence labeling methods, such as the well-known conditional random field (CRF) framework (Lafferty, McCallum, and Pereira 2001).

Denoting the word form and the corresponding label sequence as x and y, respectively, the CRFs directly model the conditional probability of the segmentation given the word form, that is, $p(y \mid x; w)$. The model parameters w are estimated discriminatively from the annotated data set $\mathcal{D}$ using iterative learning algorithms (Lafferty, McCallum, and Pereira 2001; Collins 2002). Subsequent to estimation, the CRF model segments word forms x by using maximum a posteriori (MAP) graph inference, that is, solving an optimization problem

$$z = \arg\max_{u} p\,(u \mid x; w) \tag{3}$$

using the standard Viterbi search (Lafferty, McCallum, and Pereira 2001).

As it turns out, the CRF model can learn to segment words with a surprisingly high accuracy from a relatively small $\mathcal{D}$, that is, without utilizing any of the available unannotated word forms $\mathcal{U}$. Particularly, Ruokolainen et al. (2013) showed that it is sufficient to use simple left and right substring context features that are naturally accommodated by the discriminative parameter estimation procedure. Moreover, Ruokolainen et al. (2014) showed that the CRF-based approach can be successfully extended to semi-supervised learning settings in a straightforward manner via feature set expansion by utilizing predictions of unsupervised segmentation algorithms. By utilizing this approach, the CRF model learns to associate the output of the unsupervised algorithms, such as the Morfessor and adaptor grammar methods, in relation to the surrounding substring context.

Other Work. In addition to the algorithms discussed here, there exist numerous other segmentation approaches applicable in the minimally supervised learning setting. As the earliest example of work in this line, consider obtaining segmentations using the classic letter successor variety (LSV) method of Harris (1955). The LSV method utilizes the insight that the predictability of successive letters should be high within morph segments, and low at the boundaries. Consequently, a high variety of letters following a prefix indicates a high probability of a boundary. Whereas LSV score tracks predictability given prefixes, the same idea can be utilized for suffixes, providing the letter predecessor variety (LPV) method. As for the minimally supervised learning setting, the LSV/LPV method can be used most straightforwardly by counting the LSV/LPV scores from unannotated data and, subsequently, tuning the necessary threshold values on the annotated data (Çöltekin 2010). On the other hand, one could also use the LSV/LPV values as features for a classification model, in which case the threshold values can be learned discriminatively based on the available annotated data. The latter approach is essentially realized in the event the LSV/PSV scores are provided for the CRF model discussed earlier (Ruokolainen et al. 2014).

As for more recent work, we first refer to the generative log-linear model of Poon, Cherry, and Toutanova (2009). Similarly to the Morfessor model family, this approach is based on defining a joint probability distribution over the unannotated word forms $\mathcal{U}$ and the corresponding segmentations $\mathcal{S}$. The distribution is log-linear in form and is denoted as $p(\mathcal{U}, \mathcal{S}; \theta)$, where θ is the model parameter vector. Again, similarly to the Morfessor framework, Poon, Cherry, and Toutanova (2009) learn a morph lexicon that is subsequently used to generate segmentations for new word forms. The learning is controlled using prior distributions on both corpus and lexicon, which penalize exceedingly complex morph lexicon (similarly to Morfessor) and exceedingly segmented corpus, respectively. The log-linear form of $p(\mathcal{U}, \mathcal{S}; \theta)$ enables the approach to use a wide range of overlapping features. Particularly, Poon, Cherry, and Toutanova (2009) utilize a morph-context feature set with individual features defined for each morph and morph substring contexts. In addition to unsupervised learning, they present experiments in the semi-supervised setting. Specifically, they accomplish this by fixing the segmentations of annotated words in $\mathcal{D}$, according to their gold standard segmentation. Note, however, that this approach of extending a generative model does not necessarily utilize the supervision efficiently, as discussed previously regarding the Morfessor method family.

Finally, we briefly mention a range of recently published methods (Monson, Hollingshead, and Roark 2010; Spiegler and Flach 2010; Kılıç and Bozşahin 2012; Eger 2013). The Paramor approach presented by Monson, Hollingshead, and Roark (2010) defines a rule-based system for unsupervised learning of morphological paradigms. The Promodes system of Spiegler and Flach (2010) defines a family of generative probabilistic models for recovering segment boundaries in an unsupervised fashion. The algorithm of Kılıç and Bozşahin (2012) is based on a generative hidden Markov model (HMM), in which the HMM learns to generate morph sequences for given word forms in a semi-supervised fashion. Finally, Eger (2013) presents work on fully supervised segmentation by exhaustive enumeration and a generative Markov model on morphs. As for the minimally supervised learning setting, the Paramor system learns mainly from unannotated data $\mathcal{U}$ and utilizes annotated data $\mathcal{D}$ to adjust the required threshold value. The Promodes models can be trained either in an unsupervised manner on $\mathcal{U}$ or in a supervised manner on $\mathcal{D}$. The algorithm of Kılıç and Bozşahin (2012) learns mainly from unannotated data $\mathcal{U}$ and incorporates supervision from the annotated corpus in the form of manually selected statistics: the inclusion of the statistics yields a large

improvement in performance. Lastly, in their work with the supervised enumeration approach, Eger (2013) assumes a large (on the order of tens of thousands) amount of annotated word forms available for learning. Thus, it is left for future work to determine if the approach could be applied successfully in the minimally supervised learning setting.

3.3.2 Summary. Here we aim to summarize the fundamental differences and similarities between the varying learning approaches discussed in the previous section.

Learning Lexicons versus Detecting Boundaries. We begin by dividing the methods described earlier into two—**lexicon-based** (Creutz et al. 2007; Poon, Cherry, and Toutanova 2009; Monson, Hollingshead, and Roark 2010; Kılıç and Bozşahin 2012; Eger 2013; Sirts and Goldwater 2013) and **boundary detection** (Harris 1955; Spiegler and Flach 2010; Ruokolainen et al. 2013)—categories. In the former, the model learns lexical units, whereas in the latter the model learns properties of morph boundaries. For example, in the case of Morfessor (Creutz et al. 2007) the lexical units correspond to morphs whereas in AGs (Sirts and Goldwater 2013) the units are parse trees. Meanwhile, consider the CRF approach of Ruokolainen et al. (2013) and the classical approach of Harris (1955), which identify morph boundary positions using substring contexts and letter successor varieties, respectively. In general, whether it is easier to discover morphs or morph boundaries is largely an empirical question. So far, only the method of Poon, Cherry, and Toutanova (2009) has explicitly modeled both a morph lexicon and features describing character n-grams at morpheme boundaries.

Generative versus Discriminative Learning. The second main distinction divides the models into **generative** and **discriminative** approaches. The generative approaches (Creutz et al. 2007; Poon, Cherry, and Toutanova 2009; Spiegler and Flach 2010; Monson, Hollingshead, and Roark 2010; Kılıç and Bozşahin 2012; Eger 2013; Sirts and Goldwater 2013) model the joint distribution of word forms and their corresponding segmentations, whereas discriminative (Harris 1955; Ruokolainen et al. 2013) approaches directly estimate a conditional distribution of segmentation *given* a word form. In other words, whereas generative methods generate both word forms and segmentations, the discriminative methods generate only segmentations given word forms. The generative models are naturally applicable for unsupervised learning. Meanwhile, discriminative modeling always requires some annotated data, thus excluding the possibility of unsupervised learning. Lastly, it appears that most lexicon-based methods are generative and most boundary detection methods are discriminative. However, note that this is a trend rather than a rule, as exemplified by generative boundary detection method of Spiegler and Flach (2010).

Semi-Supervised Learning Approaches. Both generative and discriminative models can be extended to utilize annotated as well as unannotated data in a semi-supervised manner. However, the applicable techniques differ. For generative models, semi-supervised learning is in principle trivial: For the labeled word forms $\mathcal{D}$, the segmentation is fixed to its correct value, as exemplified by the approaches of Poon, Cherry, and Toutanova (2009), Spiegler and Flach (2010), and Sirts and Goldwater (2013). On the other hand, the semi-supervised setting also makes it possible to apply discriminative techniques to generative models. In particular, model hyperparameters can be selected to optimize segmentation performance, rather than some generative objective, such as likelihood. Special cases of hyperparameter selection include the weighted objective function

(Kohonen, Virpioja, and Lagus 2010), data selection (Virpioja, Kohonen, and Lagus 2011; Sirts and Goldwater 2013), and grammar template selection (Sirts and Goldwater 2013). As for the weighted objective function and grammar template selection, the weights and templates are optimized to maximize segmentation accuracy. Meanwhile, data selection is based on the observation that omitting some of the training data can improve segmentation accuracy (Virpioja, Kohonen, and Lagus 2011; Sirts and Goldwater 2013).

For discriminative models, the possibly most straightforward semi-supervised learning technique is adding features derived from the unlabeled data, as exemplified by the CRF approach of Ruokolainen et al. (2014). However, discriminative, semi-supervised learning is in general a much researched field with numerous diverse techniques (Zhu and Goldberg 2009). For example, merely for the CRF model alone, there exist several proposed semi-supervised learning approaches (Jiao et al. 2006; Mann and McCallum 2008; Wang et al. 2009).

On Local Search. In what follows, we will discuss a potential pitfall of some algorithms that utilize local search procedures in the parameter estimation process, as exemplified by the Morfessor model family (Creutz et al. 2007). As discussed in Section 3.3.1, the Morfessor algorithm finds a local optimum of the objective function using a local search procedure. This complicates model development because if two model variants perform differently empirically, it is uncertain whether it is because of a truly better model or merely better fit with the utilized parameter estimation method, as discussed also by Goldwater (2006, Section 4.2.2.3). Therefore, in contrast, within the adaptor grammar framework (Johnson, Griffiths, and Goldwater 2006; Sirts and Goldwater 2013), the focus has not been on finding a single best model, but rather on finding the posterior distribution over segmentations of the words. Another approach to the problem of bad local optima is to start a local search near some known good solution. This approach is taken in Morfessor FlatCat, for which it was found that initializing the model with the segmentations produced by the supervised CRF model (with a convex objective function) yields improved results (Grönroos et al. 2014).

4. Experiments

In this section, we perform an empirical comparison of segmentation algorithms in the semi-supervised learning setting. The purpose of the presented experiments is to extend the current literature by considering a wider range of languages compared with previous work, and by providing an in-depth error analysis.

4.1 Data

We perform the experiments on four languages, namely, English, Estonian, Finnish, and Turkish. The English, Finnish, and Turkish data are from the Morpho Challenge 2009/2010 data set (Kurimo et al. 2009; Kurimo, Virpioja, and Turunen 2010). The annotated Estonian data set is acquired from a manually annotated, morphologically disambiguated corpus,[1] and the unannotated word forms are gathered from the Estonian Reference Corpus (Kaalep et al. 2010). Table 3 shows the total number of instances available for model estimation and testing.

1 Available at `http://www.cl.ut.ee/korpused/morfkorpus/index.php?lang=en`.

Table 3
Number of word types in the data sets.

	English	Estonian	Finnish	Turkish
train (unannotated)	384,903	3,908,820	2,206,719	617,298
train (annotated)	1,000	1,000	1,000	1,000
development	694	800	835	763
test	10×1,000	10×1,000	10×1,000	10×1,000

4.2 Compared Algorithms

We present a comparison of the Morfessor family (Creutz and Lagus 2002, 2005, 2007; Kohonen, Virpioja, and Lagus 2010; Grönroos et al. 2014), the adaptor grammar framework (Sirts and Goldwater 2013), and the conditional random fields (Ruokolainen et al. 2013, 2014). These methods have freely available implementations for research purposes.

The log-linear model presented by Poon, Cherry, and Toutanova (2009) is omitted because it does not have a freely available implementation. However, the model has been compared in the semi-supervised learning setting on Arabic and Hebrew with CRFs and Morfessor previously by Ruokolainen et al. (2013). In these experiments, the model was substantially outperformed on both languages by the CRF method and on Hebrew by Morfessor.

In order to provide a strong baseline for unsupervised learning results, we performed preliminary experiments using the model presented by Lee, Haghighi, and Barzilay (2011).[2] Their model learns segmentation in an unsupervised manner by exploiting syntactic context of word forms observed in running text and has shown promising results for segmentation of Arabic. In practice, we found that when using the method's default hyperparameters, it did not yield nearly as good results as the other unsupervised methods on our studied data sets. Adjusting the hyperparameters turns out to be complicated by the computational demands of the method. When utilizing the same computer set-up as for the other models, training the method requires limiting the maximum word length of analyzed words to 12 in order for the model to fit in memory, as well as requiring weeks of runtime for a single run. We decided to abandon further experimentation with the method of Lee, Haghighi, and Barzilay (2011), as optimizing its hyperparameters was computationally infeasible.

4.3 Evaluation

This section describes the utilized evaluation measures and the performed error analysis.

4.3.1 Boundary Precision, Recall, and F1-score. The word segmentations are evaluated by comparison with reference segmentations using **boundary precision**, **boundary recall**, and **boundary F1-score**. The boundary F1-score, or F1-score for short, equals the harmonic mean of precision (the percentage of correctly assigned boundaries with respect

2 Implementation is available at `http://people.csail.mit.edu/yklee/code.html`.

to all assigned boundaries) and recall (the percentage of correctly assigned boundaries with respect to the reference boundaries):

$$\text{Precision} = \frac{C(\text{correct})}{C(\text{proposed})}, \tag{4}$$

$$\text{Recall} = \frac{C(\text{correct})}{C(\text{reference})}. \tag{5}$$

We follow Virpioja et al. (2011) and use type-based macro-averages. However, we handle word forms with alternative analyses in a different fashion. Instead of penalizing algorithms that propose an incorrect number of alternative analyses, we take the best match over the alternative reference analyses (separately for precision and recall). This is because all the methods considered in the experiments provide a single segmentation per word form.

Throughout the experiments, we establish statistical significance with confidence level 0.95, according to the standard one-sided Wilcoxon signed-rank test performed on 10 random subsets of 1,000 word forms drawn from the complete test sets (subsets may contain overlapping word forms).

Because we apply a different treatment of alternative analyses, the results reported in this article are not directly comparable to the boundary F1-scores reported for the Morpho Challenge competitions (Kurimo et al. 2009; Kurimo, Virpioja, and Turunen 2010). However, the best boundary F1-scores for all languages reported in Morpho Challenge have been achieved with the semi-supervised Morfessor Baseline algorithm (Kohonen, Virpioja, and Lagus 2010), which is included in the current experiments.

4.3.2 Error Analysis. We next discuss the performed error analysis. The purpose of the error analysis is to gain a more detailed understanding into what kind of errors the methods make, and how the error types affect the overall F1-scores. To this end, we use a categorization of morphs into the categories PREFIX, STEM, and SUFFIX, in addition defining a separate category for DASH. For the English and Finnish sections of the Morpho Challenge data set, the segmentation gold standard annotation contain additional information for each morph, such as part-of-speech for stems and morphological categories for affixes, which allows us to assign each morph into one of the morph type categories. In some rare cases the tagging is not specific enough, and we choose to assign the tag UNKNOWN. However, as we are evaluating segmentations, we lack the morph category information for the proposed analyses. Consequently, we cannot apply a straightforward category evaluation metric, such as category F1-score. In what follows, we instead show how to use the categorization on the gold standard side to characterize the segmentation errors.

We first observe that errors come in two kinds, **over-segmentation** and **under-segmentation**. In over-segmentation, boundaries are incorrectly assigned within morph segments, and in under-segmentation, the segmentation fails to uncover correct morph boundaries. For example, consider the English compound word form *girlfriend* with a correct analysis *girl+friend*. Then, an under-segmentation error occurs in the event the model fails to assign a boundary between the segments *girl* and *friend*. Meanwhile, over-segmentation errors take place if any boundaries are assigned within the two compound segments *girl* and *friend*, such as *g+irl* or *fri+end*.

As for the relationship between these two error types and the precision and recall measures in Equations (4) and (5), we note that over-segmentation solely affects

precision, whereas under-segmentation only affects recall. This is evident as the measures can be written equivalently as:

$$\text{Precision} = \frac{C(\text{proposed}) - C(\text{over-segm.})}{C(\text{proposed})} = 1 - \frac{C(\text{over-segm.})}{C(\text{proposed})}, \quad (6)$$

$$\text{Recall} = \frac{C(\text{reference}) - C(\text{under-segm.})}{C(\text{reference})} = 1 - \frac{C(\text{under-segm.})}{C(\text{reference})}. \quad (7)$$

In the error analysis, we use these equivalent expressions as they allow us to examine the effect of *reduction* in precision and recall caused by over-segmentation and under-segmentation, respectively.

The over-segmentation errors occur when a segment that should remain intact is split. Thus, these errors can be assigned into categories c according to the morph tags PREFIX, STEM, SUFFIX, and UNKNOWN. The segments in the category DASH cannot be segmented and do not, therefore, contribute to over-segmentation errors. We then decompose the precision and recall reductions in Equations (6) and (7) into those caused by errors in each category indexed by c and d:

$$\text{Precision} = 1 - \sum_c \frac{C(\text{over-segm. } (c))}{C(\text{proposed})}, \quad (8)$$

$$\text{Recall} = 1 - \sum_d \frac{C(\text{under-segm. } (d))}{C(\text{reference})}. \quad (9)$$

Equation (8) holds because

$$\frac{C(\text{over-segm.})}{C(\text{reference})} = \frac{\sum_c C(\text{over-segm. } (c))}{C(\text{reference})} = \sum_c \frac{C(\text{over-segm. } (c))}{C(\text{reference})}, \quad (10)$$

where c indexes the over-segmentation error categories. The expression for recall in Equation (9) can be derived analogously, but it must be noted that the categorization d by error type differs from that of precision as each under-segmentation error occurs at a segment boundary, such as STEM-SUFFIX, STEM-STEM, PREFIX-STEM, rather than in the middle of a segment. To simplify analysis, we have grouped all segment boundaries, in which either the left or right segment category is DASH into the CONTAINS DASH category. Boundary types that occur fewer than 100 times in the test data are merged into the OTHER category.

Table 4 shows the occurrence frequency of each boundary category, averaged over alternative analyses. Evidently, we expect the total precision scores to be most influenced by over-segmentation of STEM and SUFFIX segment types because of their high frequencies. Similarly, the overall recall scores are expected to be most impacted by under-segmentation of STEM-SUFFIX and SUFFIX-SUFFIX boundaries. Finnish is also substantially influenced by the STEM-STEM boundary, indicating that Finnish uses compounding frequently.

For simplicity, when calculating the error analysis, we forgo the sampling procedure of taking $10 \times 1,000$ word forms from the test set, used for the overall F1-score for statistical significance testing by Virpioja et al. (2011). Rather, we calculate the error analysis on the union of these sampled sets. As the sampling procedure may introduce

Table 4
Absolute and relative frequencies of the boundary categories in the error analysis. The numbers are averaged over the alternative analyses in the reference annotation.

Category	English		Finnish	
STEM	38,608.8	(82.2%)	72,666.0	(81.3%)
SUFFIX	7,172.9	(15.3%)	15,384.9	(17.2%)
PREFIX	1,152.8	(2.5%)	946.5	(1.1%)
UNKNOWN	54.5	(0.1%)	414.0	(0.5%)
STEM-SUFFIX	5,349.2	(62.6%)	9,889.9	(45.8%)
SUFFIX-SUFFIX	1,481.0	(17.3%)	5,917.5	(27.4%)
STEM-STEM	613.4	(7.2%)	3,538.0	(16.4%)
SUFFIX-STEM	n/a	n/a	1,501.0	(6.9%)
CONTAINS DASH	458.0	(6.5%)	426.0	(2.0%)
PREFIX-STEM	554.3	(5.4%)	235.2	(1.1%)
OTHER	91.0	(1.1%)	105.4	(0.5%)

the same word form in several samples, the error analysis precisions and recalls are not necessarily identical to the ones reported for the overall results.

In summary, although we cannot apply category F1-scores, we can instead categorize each error by type. These categories then map directly to either reduced precision or recall. Interpreting precision and recall requires some care as it is always possible to reduce over-segmentation errors by segmenting less and, conversely, to reduce under-segmentation errors by segmenting more. However, if this is taken into account, the error categorization can be quite informative.

4.4 Model Learning and Implementation Specifics

4.4.1 Morfessor. We use a recently released Python implementation of the Morfessor method (Virpioja et al. 2013; Smit et al. 2014).[3] The package implements both the unsupervised and semi-supervised Morfessor Baseline (Creutz and Lagus 2002, 2007; Kohonen, Virpioja, and Lagus 2010). For Morfessor FlatCat we apply the Python implementation by Grönroos et al. (2014).[4]

In its original formulation, the unsupervised Morfessor Baseline uses no hyperparameters. However, it was found by Virpioja, Kohonen, and Lagus (2011) that performance does not improve consitently with growing data because the method segments less on average for each added training word form. Therefore, we optimize the training data size by including only the most frequent words in the following sizes: 10k, 20k, 30k, 40k, 50k, 100k, 200k, 400k, ..., as well as the full set. We then choose the model yielding highest F1-score on the development set.

As for semi-supervised training of Morfessor Baseline, we perform a grid search on the development set for the hyperparameter β (see Section 3.3.1). For each value of β we use the automatic adaptation of the hyperparameter α provided by the implementation. The automatic adaptation procedure is applied during model training and is, therefore, computationally less demanding compared with grid search. Intuitively, the adaptation

3 Available at `https://github.com/aalto-speech/morfessor`.
4 Available at `https://github.com/aalto-speech/flatcat`.

functions as follows. The hyperparameter α affects how much the method segments on average. Although optimizing it for segmentation performance during training is non-trivial, one can instead apply the heuristic that the method should neither over-segment nor under-segment. Therefore, the implementation adjusts α such that the development set precision and recall become approximately equal.

In the semi-supervised training for Morfessor FlatCat, the segmentations are initialized to the ones produced by the supervised CRF model trained with the same amount of labeled training data. As automatic adaptation of the hyperparameter α has not yet been implemented for Morfessor FlatCat, values for both α and β are found by a combined grid search on the development set. The computational demands of the grid search were reduced by using the optimal hyperparameter values for Morfessor Baseline as an initial guess when constructing the grid. We also choose the non-morpheme removal heuristics used by Morfessor FlatCat for each language separately using the development set. For English, Estonian, and Finnish the heuristics described by Grönroos et al. (2014) are beneficial, but they do not fit Turkish morphology as well. For Turkish we convert non-morphemes into suffixes or stems, without modifying the segmentation.

4.4.2 Adaptor Grammars. The technical details of the AG model are described by Johnson, Griffiths, and Goldwater (2006) and the inference details are described by Johnson, Griffiths, and Goldwater (2007). For unsupervised AG learning, we used the freely available implementation,[5] which was also the basis for the semi-supervised implementation. Table label resampling was turned on and all hyperparameters were inferred automatically as described by Johnson and Goldwater (2009). The metagrammar for AG Select is the same as described by Sirts and Goldwater (2013). Inductive learning with the posterior grammar was done with a freely available CKY parser.[6] For both unsupervised and semisupervised AG, we use a three-level collocation-submorph grammar in which the final segmentation is parsed out as a sequence of Morphs:

$$\text{Word} \rightarrow \underline{\text{Colloc}}^{+}$$

$$\underline{\text{Colloc}} \rightarrow \underline{\text{Morph}}^{+}$$

$$\underline{\text{Morph}} \rightarrow \underline{\text{SubMorph}}^{+}$$

$$\underline{\text{SubMorph}} \rightarrow \text{Char}^{+}$$

We experimented with two types of grammars, where the Word non-terminal is either cached or not. These two grammar versions have no difference when trained transductively. However, when training an inductive model, it may be beneficial to store the subtrees corresponding to whole words because these trees can be used to parse the words in the test set that were seen during training with a single rule. All models, both unsupervised and semi-supervised, are trained on 50k most frequent word types. For semi-supervised experiments, we upweight the labeled data by an integer number of times by repeatedly caching the subtrees corresponding to morphemes in the annotated data. The additional cached subtrees are rooted in the Morph non-terminal. Similarly to semi-supervised Morfessor, we experimented with initializing the segmentations with

5 Available at `http://web.science.mq.edu.au/~mjohnson/Software.htm`.
6 Also obtained from `http://web.science.mq.edu.au/~mjohnson/Software.htm`.

the output of the supervised CRF model, which in some cases resulted in improved accuracy over the random initialization. We searched the optimal values for each experiment for the upweighting factor, cached versus non-cached root non-terminal, and random versus CRF initialization on the development set.

An AG model is stochastic and each segmentation result is just a single sample from the posterior. A common approach in such a case is to take several samples and report the average result. Maximum marginal decoding (MMD) (Johnson and Goldwater 2009; Stallard et al. 2012) that constructs a marginal distribution from several independent samples and returns their mean value has been shown to improve the sampling-based models' results about 1–2 percentage points. Although the AG model uses sampling for training, the MMD is not applicable here because during test time the segmentations are obtained using parsing. However, we propose another way of achieving the gain in a similar range to the MMD. We train five different models and concatenate their posterior grammars into a single joint grammar, which is then used as the final model to decode the test data. Our experiments show that the posterior grammar concatentation, similarly to the MMD, leads to consistent improvements of 1–2 percentage points over the mean of the individual samples.

4.4.3 CRFs. The utilized Python implementation of the CRF model follows the presentation of Ruokolainen et al. (2013, 2014).[7] As for the left and right substring features incorporated in the model, we include all substrings that occur in the training data. The maximum substring length and averaged perceptron learning of CRF model parameters are optimized on the held-out development sets following Ruokolainen et al. (2013). For semi-supervised learning, we utilize log-normalized successor and predecessor variety scores and binary Morfessor Baseline and AG features following the presentation of Ruokolainen et al. (2014). The unsupervised Morfessor Baseline and AG models are optimized on the development set as described earlier. The successor and predecessor variety scores are estimated from all the available unannotated word forms apart from words with a corpus frequency of one. The count cutoff is applied as a means of noise reduction by removing peripheral phenomena, such as misspellings.

4.5 Results

Here we summarize the results obtained using our experiment set-up. We present overall segmentation accuracies and error analysis in Sections 4.5.1 and 4.5.2, respectively. We then discuss the results in Section 4.6.

4.5.1 Boundary Precisions, Recalls, and F1-scores. In what follows, we first review unsupervised and supervised results and, subsequently, assess the semi-supervised results.

Segmentation accuracies using unsupervised and supervised methods are presented in Table 5. As for the supervised learning using the CRF model, we report segmentation accuracies obtained using 100 and 1,000 annotated word forms. Evidently, utilizing annotated data provides a distinct advantage over learning from unannotated data. Particularly, learning the supervised CRFs using 1,000 annotated word forms results in substantially higher segmentation accuracies compared with learning in an unsupervised manner from hundreds of thousands or millions of word forms. In fact, using merely 100 annotated instances results in higher accuracies in English and Turkish

7 Available at `http://users.ics.aalto.fi/tpruokol/`.

Table 5
Precision, recall, and F1-scores for unsupervised and supervised methods.

Method	Train (ann.)	Train (unann.)	Pre.	Rec.	F1
English					
MORFESSOR BASELINE (USV)	0	384,903	76.3	76.3	76.3
AG (USV)	0	384,903	62.2	84.4	71.7
CRF (SV)	100	0	86.0	72.7	78.8
CRF (SV)	1,000	0	91.6	81.2	86.1
Estonian					
MORFESSOR BASELINE (USV)	0	3,908,820	76.4	70.4	73.3
AG (USV)	0	3,908,820	78.4	73.4	75.8
CRF (SV)	100	0	79.2	59.1	67.7
CRF (SV)	1,000	0	88.4	76.7	82.1
Finnish					
MORFESSOR BASELINE (USV)	0	2,206,719	70.2	51.9	59.7
AG (USV)	0	2,206,719	68.1	68.1	68.1
CRF (SV)	100	0	73.0	59.4	65.5
CRF (SV)	1,000	0	88.3	79.7	83.8
Turkish					
MORFESSOR BASELINE (USV)	0	617,298	67.9	65.8	66.8
AG (USV)	0	617,298	72.7	76.5	74.6
CRF (SV)	100	0	84.6	71.8	77.7
CRF (SV)	1,000	0	90.0	87.3	88.6

The columns titled *Train (unann.)* denote the number of unannotated word forms utilized in learning. The columns titled *Train (ann.)* denote the number of annotated word forms.

compared with the unsupervised methods. The balance between precision and recall can be analyzed to assess how well the different methods are tuned to the amount of segmentation present in the gold standard. As discussed in Section 4.3.2, high precision in combination with low recall indicates under-segmentation, whereas high recall and low precision indicates over-segmentation. Morfessor appears to favor precision over recall (see Finnish) in the event a trade-off takes place. In contrast, the AG heavily favors recall (see English). Meanwhile, the supervised CRF model consistently prefers higher precision over recall.

These unsupervised and supervised learning results utilize the available data only partially. Thus, we next discuss results obtained using semi-supervised learning—that is, when utilizing all available annotated and unannotated word forms. The obtained segmentation accuracies are presented in Table 6. We summarize the results as follows. First, the semi-supervised CRF approach CRF (SSV) yielded highest segmentation accuracies for all considered languages and data set sizes. The improvements over other models are statistically significant. Compared with the supervised CRF model, the semi-supervised extension successfully increases the recall while maintaining the high precision. As for the Morfessor family, MORF.FC (SSV) yields significantly higher F1-scores compared with MORF.BL (SSV) on all languages. However, we found that without the CRF initialization of MORF.FC (SSV), the performance gap decreases substantially (cf. similar results reported by Grönroos et al. [2014]). On the other hand, the variants appear to behave in a similar manner in that, in the majority of cases, both approaches increase the obtained precision and recall in a balanced manner compared with the unsupervised approach MORF. BL (USV). Meanwhile, the AG variants AG (SSV) and

Table 6
Precision, recall, and F1-scores for semi-supervised methods.

Method	Train (ann.)	Train (unann.)	Pre.	Rec.	F1
English					
MORFESSOR BASELINE (SSV)	100	384,903	81.7	82.8	82.2
MORFESSOR FLATCAT (SSV)	100	384,903	83.6	83.0	83.3
AG (SSV)	100	384,903	69.0	85.8	76.5
AG SELECT (SSV)	100	384,903	75.9	79.4	77.6
CRF (SSV)	100	384,903	87.6	81.0	**84.2**
MORFESSOR BASELINE (SSV)	1,000	384,903	84.4	83.9	84.1
MORFESSOR FLATCAT (SSV)	1,000	384,903	86.9	85.2	86.0
AG (SSV)	1,000	384,903	69.8	87.1	77.5
AG SELECT (SSV)	1,000	384,903	76.7	82.3	79.4
CRF (SSV)	1,000	384,903	89.3	87.0	**88.1**
Estonian					
MORFESSOR BASELINE (SSV)	100	3,908,820	77.0	76.1	76.5
MORFESSOR FLATCAT (SSV)	100	3,908,820	81.8	74.5	77.9
AG (SSV)	100	3,908,820	71.8	75.5	73.6
AG SELECT (SSV)	100	3,908,820	60.9	90.4	72.8
CRF (SSV)	100	3,908,820	81.5	82.1	**81.8**
MORFESSOR BASELINE (SSV)	1,000	3,908,820	80.6	80.7	80.7
MORFESSOR FLATCAT (SSV)	1,000	3,908,820	84.7	82.0	83.3
AG (SSV)	1,000	3,908,820	67.1	88.8	76.4
AG SELECT (SSV)	1,000	3,908,820	62.8	90.3	74.1
CRF (SSV)	1,000	3,908,820	90.2	86.3	**88.2**
Finnish					
MORFESSOR BASELINE (SSV)	100	2,206,719	69.8	70.8	70.3
MORFESSOR FLATCAT (SSV)	100	2,206,719	77.6	73.6	75.5
AG (SSV)	100	2,206,719	65.5	70.5	67.9
AG SELECT (SSV)	100	2,206,719	66.8	73.6	70.0
CRF (SSV)	100	2,206,719	80.0	77.4	**78.7**
MORFESSOR BASELINE (SSV)	1,000	2,206,719	76.0	78.0	77.0
MORFESSOR FLATCAT (SSV)	1,000	2,206,719	81.6	80.2	80.9
AG (SSV)	1,000	2,206,719	69.7	77.6	73.4
AG SELECT (SSV)	1,000	2,206,719	69.4	74.3	71.8
CRF (SSV)	1,000	2,206,719	89.3	87.9	**88.6**
Turkish					
MORFESSOR BASELINE (SSV)	100	617,298	76.6	80.5	78.5
MORFESSOR FLATCAT (SSV)	100	617,298	80.2	83.9	82.0
AG (SSV)	100	617,298	74.1	82.8	78.2
AG SELECT (SSV)	100	617,298	69.0	82.3	75.0
CRF (SSV)	100	617,298	81.3	86.0	**83.5**
MORFESSOR BASELINE (SSV)	1,000	617,298	85.1	89.4	87.2
MORFESSOR FLATCAT (SSV)	1,000	617,298	84.9	92.2	88.4
AG (SSV)	1,000	617,298	77.0	90.9	83.4
AG SELECT (SSV)	1,000	617,298	70.5	80.4	75.1
CRF (SSV)	1,000	617,298	89.3	92.0	**90.7**

AG SELECT (SSV) heavily favor recall over precision, indicating over-segmentation.[8] Lastly, in contrast with the unsupervised learning results, in the semi-supervised setting the AG framework is significantly outperformed by the Morfessor variants.

4.5.2 Error Analysis. Next, we examine how different error types contribute to the obtained precision and recall measures, and, consequently, the overall F1-scores. To this end, we discuss the error analyses for English and Finnish presented in Tables 7 and 8, respectively.

Baselines. The first two lines in Tables 7 and 8 present the baseline models WORDS and LETTERS. The WORDS model corresponds to an approach in which no segmentation is performed, that is, all the word forms are kept intact. The LETTERS approach assigns a segment boundary between all adjacent letters. These approaches maximize precision (WORDS) and recall (LETTERS) at the cost of the other. In other words, no model can produce more over-segmentation errors compared with LETTERS, and no model can produce more under-segmentation errors compared with WORDS.[9]

Given the baseline results, we observe that the overall precision scores are most influenced by over-segmentation of STEM and SUFFIX segment types because of their high frequencies. Similarly, the overall recall scores are most impacted by under-segmentation of STEM-SUFFIX and SUFFIX-SUFFIX boundaries. Finnish recall is also substantially influenced by the STEM-STEM boundary, indicating that Finnish uses compounding frequently.

Morfessor. Similarly to the baseline (WORDS and LETTERS) results, the majority of over-segmentation errors yielded by the Morfessor variants take place within the STEM and SUFFIX segments, and most under-segmentation errors occur at the STEM-SUFFIX and SUFFIX-SUFFIX boundaries. When shifting from unsupervised learning using MORF.BL (USV) to semi-supervised learning using MORF.BL (SSV) and MORF.FC (SSV), the over-segmentation problems are alleviated rather substantially, resulting in higher overall precision scores. For example, consider the word form *countermanded*, for which MORF.BL (SSV) assigns the correct segmentation *countermand+ed*, but which is severely oversegmented by MORF.BL (USV) as *counter+man+d+ed*. One also observes a dramatic increase in the overall recall scores, indicating a smaller amount of under-segmentation taking place. For example, consider the word form *products*, for which MORF.BL (SSV) assigns the correct segmentation *product+s*, but for which MORF.BL (USV) assigns no boundaries. However, the under-segmentation errors do not decrease consistently: Although the STEM-SUFFIX and SUFFIX-SUFFIX errors are decreased substantially, one additionally observes a decline or no change in the model's ability to uncover STEM-STEM and PREFIX-STEM boundaries.

8 Generally, in the presence of annotated training data, under-segmentation and over-segmentation can be avoided by explicitly tuning the average level of segmentation. Such tuning is performed for Morfessor with the weighted objective function and for AG by choosing the level in the parse tree from which to extract the segmentations. By default, the AG segmentations were extracted from the Morph level as this gave the highest score on the development set. However, the Estonian segmentations are extracted from the Colloc level, which also explains why in the Estonian case the precision is higher than recall. These results suggest that AG (SSV) may benefit from yet another layer in the grammar, which would help to learn a better balance between precision and recall.

9 Intuitively, WORDS should yield zero recall. However, when applying macro averaging, a word having a gold standard analysis with no boundaries yields a zero denominator and is therefore undefined. To correct for this, we interpret such words as having recall 1, which explains the non-zero recall for WORDS.

Table 7
Error analysis for English.

	Over-Segmentation					Under-Segmentation						
Method	STEM	SUFFIX	PREFIX	UNKNOWN	**PRE / TOTAL**	STEM-SUFFIX	SUFFIX-SUFFIX	STEM-STEM	PREFIX-STEM	CONTAINS DASH	OTHER	**REC / TOTAL**
WORDS	0.0	0.0	0.0	0.0	100.0	55.1	8.6	5.9	4.4	2.5	0.6	23.1
LETTERS	71.1	11.8	1.7	0.3	15.1	0.0	0.0	0.0	0.0	0.0	0.0	100.0
MORF.BL (USV)	20.6	2.9	0.0	0.1	76.4	17.0	4.7	0.6	1.1	0.0	0.2	76.4
MORF.BL (SSV)	14.3	1.3	0.1	0.0	84.4	9.8	0.6	2.8	2.1	0.0	0.4	84.3
MORF.FC (SSV)	11.2	1.7	0.0	0.1	87.1	8.6	0.5	2.2	2.5	0.1	0.4	85.5
AG (USV)	31.8	5.8	0.1	0.0	62.3	10.6	3.5	0.1	0.7	0.2	0.2	84.7
AG (SSV)	27.8	2.1	0.1	0.1	70.0	10.1	1.4	0.2	0.6	0.2	0.2	87.3
AG SELECT (SSV)	18.4	4.8	0.0	0.1	76.6	8.2	1.4	2.2	4.1	1.5	0.4	82.2
CRF (SV)	7.3	0.9	0.1	0.0	91.8	10.4	0.5	4.2	2.9	0.1	0.4	81.5
CRF (SSV)	9.6	0.8	0.0	0.1	89.5	8.4	0.5	1.4	1.9	0.0	0.4	87.4

Over-segmentation and under-segmentation errors reduce precision and recall, respectively. For example, the total precision of MORF. BL (USV) is obtained as $100.0 - 20.6 - 2.9 - 0.0 - 0.1 = 76.4$. The lines MORF. BL (USV), MORF. BL (SSV), and MORF. FC (SSV) correspond to the unsupervised Morfessor Baseline, semi-supervised Morfessor Baseline, and semi-supervised Morfessor FlatCat models, respectively.

Table 8
Error analysis for Finnish.

	Over-Segmentation					Under-Segmentation							
Method	STEM	SUFFIX	PREFIX	UNKNOWN	**PRE / TOTAL**	STEM-SUFFIX	SUFFIX-SUFFIX	STEM-STEM	SUFFIX-STEM	CONTAINS DASH	PREFIX-STEM	OTHER	**REC / TOTAL**
WORDS	0.0	0.0	0.0	0.0	100.0	49.2	21.8	17.2	4.8	1.4	1.0	0.6	4.1
LETTERS	65.2	13.8	0.7	0.6	19.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
MORF.BL (USV)	26.6	3.4	0.0	0.2	69.7	28.8	17.1	1.7	0.5	0.0	0.1	0.2	51.6
MORF.BL (SSV)	20.8	2.9	0.0	0.2	76.1	13.6	5.9	1.9	0.5	0.0	0.1	0.1	78.0
MORF.FC (SSV)	15.3	2.9	0.0	0.1	81.7	12.2	5.2	1.5	0.6	0.1	0.1	0.1	80.2
AG (USV)	28.3	3.3	0.1	0.2	68.1	19.0	11.5	0.7	0.2	0.3	0.0	0.2	68.1
AG (SSV)	27.9	2.1	0.1	0.2	69.7	14.7	6.5	0.7	0.2	0.1	0.1	0.1	77.6
AG SELECT (SSV)	24.2	6.1	0.0	0.1	69.5	13.2	7.8	2.4	1.1	0.8	0.2	0.1	74.4
CRF (SV)	9.3	2.3	0.0	0.0	88.3	10.7	2.2	5.8	1.1	0.1	0.3	0.2	79.7
CRF (SSV)	9.2	1.4	0.0	0.1	89.3	8.0	2.3	1.2	0.4	0.1	0.1	0.2	87.8

Over-segmentation and under-segmentation errors reduce precision and recall, respectively. The lines MORF. BL (USV), MORF. BL (SSV), and MORF. FC (SSV) correspond to the unsupervised Morfessor Baseline, the semi-supervised Morfessor Baseline, and semi-supervised Morfessor FlatCat models, respectively.

Adaptor Grammars. Similarly to the baseline and Morfessor results, the majority of over-segmentation errors yielded by the AG variants occur within the STEM and SUFFIX segments. Compared with the unsupervised AG (USV) approach, the first semi-supervised extension AG (SSV) manages to reduce over-segmentation of the STEM segments slightly and SUFFIX segments substantially, thus resulting in overall higher precision. Meanwhile, the second extension AG SELECT (SSV) also results in overall higher precision by reducing over-segmentation of STEM segments substantially—although, for Finnish, SUFFIX is oversegmented compared with AG SELECT (USV). On the other hand, whereas both AG (SSV) and AG SELECT (SSV) improve recall on Finnish compared to AG (USV), only AG (SSV) succeeds in improving recall for English. This is because the AG SELECT (SSV) variant decreases the model's ability to capture other than STEM-SUFFIX and SUFFIX-SUFFIX boundaries compared with the unsupervised AG (USV) approach.

Conditional Random Fields. In contrast to the Morfessor and AG frameworks, the error patterns produced by the CRF approach do not directly follow the baseline approaches. Particularly, we note that the supervised CRF (SV) approach successfully captures SUFFIX-SUFFIX boundaries and fails to find STEM-STEM boundaries—that is, behaves in an opposite manner compared with the baseline results. CRF (SV) also under-segments the less-frequent PREFIX-STEM and STEM-SUFFIX boundaries for English and Finnish, respectively. Meanwhile, the semi-supervised extension CRF (SSV) alleviates the problem of finding STEM-STEM boundaries substantially, resulting in improvement in overall recall. For example, CRF (SSV) correctly segments compound forms *rainstorm* and *windpipe* as *rain+storm* and *wind+pipe*, whereas CRF (SV) incorrectly assigns no segmentation boundaries to either of these forms. Note that improving recall means that CRF (SSV) is required to segment more compared with CRF (SV). For English, this increased segmentation results in a slight increase in over-segmentation of STEM—that is, the model trades off the increase in recall for precision. For example, whereas CRF (SV) correctly segments *ledgers* as *ledger+s*, CRF (SSV) yields an incorrect segmentation *led+ger+s*.

4.6 Discussion

When increasing the amount of data utilized for learning—that is, when shifting from fully unsupervised or supervised learning to semi-supervised learning—we naturally expect the segmentation method families to improve their performance measured using the F1-score. Indeed, as shown in Tables 5 and 6, this improvement takes place within all considered approaches. In some cases, as exemplified by the CRF model on English, achieving a higher F1-score may require a trade-off between precision and recall—that is, the model lowers precision somewhat to gain recall (or vice versa). However, by examining the error analyses in Tables 7 and 8, we also observe the occurrence of a second kind of trade-off, in which the semi-supervised Morfessor and AG approaches trade off under-segmentation errors to other under-segmentation errors. Particularly, although the STEM-SUFFIX and SUFFIX-SUFFIX boundary recall errors are decreased, one also observes an increase in the errors at STEM-STEM and PREFIX-STEM boundaries. This type of behavior indicates an inherent inefficiency in the models' ability to utilize increasing amounts of data.

Next, we discuss potential explanations for the empirical success of the discriminatively trained CRF approach. First, discriminative training has the advantage of directly optimizing segmentation accuracy with few assumptions about the data generating process. Meanwhile, generative models can be expected to perform well only if the model

definition matches the data-generating process adequately. In general, discriminative approaches should generalize well under the condition that a sufficient amount of training data is available. Given the empirical results, this condition appears to be fulfilled for morphological segmentation in the minimally supervised setting. Second, the CRFs aim to detect boundary positions based on rich features describing substring contexts. Because the substrings are more frequent than lexical units, their use enables more efficient utilization of sparse data. For example, consider a training data that consists of a single labeled word form *kato+lla* (*on roof*). When segmenting an unseen word form *matolle* (*onto rug*), with the correct segmentation *mato+lle*, the CRFs can utilize the familiar left and right substrings *ato* and *ll*, respectively. In contrast, a lexicon-based model has a lexicon of two morphs {*kato, lla*}, neither of which match any substring of *matolle*.

Finally, we discuss how the varying approaches differ when learning to split affixes and compounds. To this end we first point out that, in the examined English and Finnish corpora, the suffix class is **closed** and has only a small number of morphemes compared with the **open** prefix and stem categories. In consequence, a large coverage of suffixes should be achievable already with a relatively small annotated data set. This observation is supported by the evident success of the fully supervised CRF method in learning suffix splitting for both considered languages. On the other hand, although more efficient at learning suffix splitting, the supervised CRF approach is apparently poor at detecting compound boundaries. Intuitively, learning compound splitting in a supervised manner seems infeasible because the majority of stem forms are simply not present in the available small annotated data set. Meanwhile, the semi-supervised CRF extension and the generative Morfessor and AG families, which do utilize the large unannotated word lists, capture the compound boundaries with an appealing high accuracy. This result again supports the intuition that in order to learn the open categories, one is required to utilize large amounts of word forms for learning. However, it appears that the necessary information can be extracted from unannotated word forms.

5. Future Work

In this section, we discuss our findings on potentially fruitful directions for future research.

5.1 On Improving Existing Approaches

Interestingly, the CRF-based segmentation method achieves its success using minimalistic, language-independent features with a simple, feature-based, semi-supervised learning extension. Therefore, it seems plausible that one could boost the accuracy further by designing richer, language-dependent feature extraction schemes. For example, one could potentially exploit features capturing vowel harmony present in Finnish, Estonian, and Turkish. As for semi-supervised learning, one can utilize unannotated word lists in a straightforward manner by using the feature set expansion approach as discussed by Ruokolainen et al. (2014). Similar expansion schemes for CRFs have also been successfully applied in the related tasks of Chinese word segmentation (Sun and Xu 2011; Wang et al. 2011) and chunking (Turian, Ratinov, and Bengio 2010). Nevertheless, there exist numerous other approaches proposed for semi-supervised learning of CRFs (Jiao et al. 2006; Mann and McCallum 2008; Wang et al. 2009) that could potentially provide an advantage over the feature-based, semi-supervised learning approach. Naturally, one could also examine utilizing these techniques simultaneously with the expanded feature sets.

As discussed in Section 3.3, it is possible for the generative models to utilize annotated data in a straightforward manner by fixing samples to their true values. This approach was taken by Poon, Cherry, and Toutanova (2009), Spiegler and Flach (2010), and Sirts and Goldwater (2013). On the other hand, as discussed in Section 3.3.1, for the Morfessor family the fixing approach was outperformed by the weighted objective function (Kohonen, Virpioja, and Lagus 2010). It has been shown that the weighting can compensate for a mismatch between the model and the data generating process (Cozman et al. 2003; Cozman and Cohen 2006; Fox-Roberts and Rosten 2014). Therefore, it would appear to be advantageous to study weighting schemes in combination with all the discussed generative models.

5.2 On Potential Novel Approaches

Based on the literature survey presented in Section 3.3.2, one can observe that there exists substantial work on generative lexicon-based approaches and methods based on discriminative boundary detection. In contrast, there exists little to no research on models utilizing lexicons and discriminative learning or generative boundary-detection approaches. In addition, as mentioned in Section 3.3.2, so far there has been little work discussing a combination of lexicon-based and boundary detection approaches. It could be fruitful to explore these modeling aspects further in the future.

6. Conclusions

We presented a comparative study on data-driven morphological segmentation in a minimally supervised learning setting. In this setting the segmentation models are estimated based on a small amount of manually annotated word forms and a large set of unannotated word forms. In addition to providing a literature survey on published methods, we presented an in-depth empirical comparison on three diverse model families. The purpose of this work is to extend the existing literature with a summarizing study on the published methodology as a whole.

Based on the literature survey, we concluded that the existing methodology contains substantial work on generative lexicon-based approaches and methods based on discriminative boundary detection. As for which approach has been more successful, both the previous work and the empirical evaluation presented here strongly imply that the current state of the art is yielded by the discriminative boundary detection methodology. In general, our analysis suggested that the models based on generative lexicon learning are inefficient at utilizing growing amounts of available data. Meanwhile, the studied discriminative boundary detection method based on the CRF framework was successful in gaining consistent reduction in all error types, given increasing amounts of data. Lastly, there exists little to no research on models utilizing lexicons and discriminative learning or generative boundary-detection approaches. Studying these directions could be of interest in future work.

Acknowledgments

This work was financially supported by Langnet (Finnish doctoral programme in language studies), the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant no. 251170) and LASTU Programme (grants no. 256887 and 259934), project *Multimodally grounded language technology* (grant no. 254104), and the Tiger University program of the Estonian Information Technology Foundation for Education.

References

Brent, Michael R. 1999. An efficient, probabilistically sound algorithm for

segmentation and word discovery. *Machine Learning*, 34(1–3):71–105.

Chrupala, Grzegorz, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2362–2367, Marrakech.

Collins, Michael. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, volume 10, pages 1–8, Philadelphia, PA.

Çöltekin, Çağrı. 2010. Improving successor variety for morphological segmentation. *LOT Occasional Series*, 16:13–28.

Cozman, Fabio and Ira Cohen. 2006. Risks of semi-supervised learning: How unlabelled data can degrade performance of generative classifiers. In Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, editors, *Semi-supervised learning*, pages 57–72. MIT press.

Cozman, Fabio Gagliardi, Ira Cohen, and Marcelo Cesar Cirelo. 2003. Semi-supervised learning of mixture models. In *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, pages 99–106, Washington, DC.

Creutz, Mathias, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1):3:1–3:29.

Creutz, Mathias and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL 2002*, pages 21–30, Philadelphia, PA.

Creutz, Mathias and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology.

Creutz, Mathias and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):1–27.

de Gispert, Adrià, Sami Virpioja, Mikko Kurimo, and William Byrne. 2009. Minimum Bayes risk combination of translation hypotheses from alternative morphological decompositions. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 73–76, Boulder, CO.

Eger, Steffen. 2013. Sequence segmentation by enumeration: An exploration. *The Prague Bulletin of Mathematical Linguistics*, 100:113–131.

Fox-Roberts, Patrick and Edward Rosten. 2014. Unbiased generative semi-supervised learning. *Journal of Machine Learning Research*, 15(1):367–443.

Goldwater, Sharon. 2006. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Green, Spence and John DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 146–155, Jeju Island.

Grönroos, Stig-Arne, 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 the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1177–1185, Dublin.

Hammarström, Harald and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Harris, Zellig S. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Hirsimäki, Teemu, Mathias Creutz, Vesa Siivola, Mikko Kurimo, Sami Virpioja, and Janne Pylkkönen. 2006. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech and Language*, 20(4):515–541.

Jiao, Feng, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*, pages 209–216, Sidney.

Johnson, Howard and Joel Martin. 2003. Unsupervised learning of morphology for English and Inuktitut. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL HLT 2003)*, pages 43–45, Edmonton.

Johnson, Mark. 2008. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the 10th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology (SIGMORPHON 2008)*, pages 20–27, Columbus, OH.

Johnson, Mark and Katherine Demuth. 2010. Unsupervised phonemic Chinese word segmentation using Adaptor Grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 528–536, Beijing.

Johnson, Mark and Sharon Goldwater. 2009. Improving nonparameteric Bayesian inference: Experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 317–325, Boulder, CO.

Johnson, Mark, Thomas L. Griffiths, and Sharon Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, pages 641–648, Vancouver.

Johnson, Mark, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2007)*, pages 139–146, Rochester, NY.

Kaalep, Heiki-Jaan, Kadri Muischnek, Kristel Uiboaed, and Kaarel Veskis. 2010. The Estonian reference corpus: Its composition and morphology-aware user interface. In *Proceedings of the 2010 Conference on Human Language Technologies – The Baltic Perspective: Proceedings of the Fourth International Conference Baltic (HLT 2010)*, pages 143–146, Riga.

Kılıç, Özkan and Cem Bozşahin. 2012. Semi-supervised morpheme segmentation without morphological analysis. In *Proceedings of the LREC 2012 Workshop on Language Resources and Technologies for Turkic Languages*, pages 52–56, Istanbul.

Kohonen, Oskar, 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 (SIGMORPHON 2010)*, pages 78–86, Uppsala.

Kurimo, Mikko, Sami Virpioja, and Ville Turunen. 2010. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo.

Kurimo, Mikko, Sami Virpioja, Ville Turunen, Graeme W. Blackwood, and William Byrne. 2009. Overview and results of Morpho Challenge 2009. In *Working Notes for the CLEF 2009 Workshop*, pages 578–597, Corfu.

Lafferty, John, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williamstown, MA.

Lee, Yoong Keok, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*, pages 1–9, Portland, OR.

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

Luong, Minh-Thang, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013)*, pages 29–37, Sofia.

Mann, G. and A. McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of the 46th Annual Meeting of Association for Computational Linguistics: Human Language Technologies (ACL HLT 2008)*, pages 870–878, Columbus, OH.

Monson, Christian, Kristy Hollingshead, and Brian Roark. 2010. Simulating morphological analyzers with stochastic taggers for confidence estimation. In *Multilingual Information Access Evaluation*

I - Text Retrieval Experiments, volume 6241 of *Lecture Notes in Computer Science*.

Narasimhan, Karthik, Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 880–885, Doha.

Neuvel, Sylvain and Sean A. Fulop. 2002. Unsupervised learning of morphology without morphemes. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON 2002)*, pages 31–40, Philadelphia, PA.

Nigam, Kamal, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134.

Pirinen, Tommi. 2008. Automatic finite state morphological analysis of Finnish language using open source resources [in Finnish]. Master's thesis, University of Helsinki.

Poon, Hoifung, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 209–217, Boulder, CO.

Qiu, Siyu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 141–150, Dublin.

Rissanen, Jorma. 1989. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Series in Computer Science, Singapore.

Ruokolainen, Teemu, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013)*, pages 29–37, Sofia.

Ruokolainen, Teemu, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 84–89. Gothenburg.

Schone, Patrick and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies (NAACL 2001)*, pages 1–9, Pittsburgh, PA.

Sirts, Kairit and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1(May):255–266.

Smit, Peter, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 21–24, Gothenburg.

Spiegler, Sebastian and Peter A. Flach. 2010. Enhanced word decomposition by calibrating the decision threshold of probabilistic models and using a model ensemble. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 375–383, Uppsala.

Stallard, David, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. 2012. Unsupervised morphology rivals supervised morphology for Arabic MT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 322–327, Jeju Island.

Sun, Weiwei and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 970–979, Edinburgh.

Turian, Joseph, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 384–394, Uppsala.

Turunen, Ville and Mikko Kurimo. 2011. Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition,

and retrieval. *ACM Transactions on Speech and Language Processing*, 8(1):1:1–1:25.
Virpioja, Sami, Oskar Kohonen, and Krista Lagus. 2010. Unsupervised morpheme analysis with Allomorfessor. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009)*, pages 609–616, Corfu.
Virpioja, Sami, Oskar Kohonen, and Krista Lagus. 2011. Evaluating the effect of word frequencies in a probabilistic generative model of morphology. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 230–237, Riga.
Virpioja, Sami, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University, Helsinki, Finland.
Virpioja, Sami, Ville Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.
Wang, Yang, Gholamreza Haffari, Shaojun Wang, and Greg Mori. 2009. A rate distortion approach for semi-supervised conditional random fields. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2008–2016, Vancouver.
Wang, Yiou, Yoshimasa Tsuruoka Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai.
Yarowsky, David and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL 2000)*, pages 207–216, Hong Kong.
Zhu, Xiaojin and Andrew B. Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130.