# Tree Structured Dirichlet Processes for Hierarchical Morphological Segmentation

Burcu Can
Hacettepe University
Department of Computer Engineering
burcucan@cs.hacettepe.edu.tr

Suresh Manandhar
University of York
Department of Computer Science
suresh@cs.york.ac.uk



*This article presents a probabilistic hierarchical clustering model for morphological segmentation. In contrast to existing approaches to morphology learning, our method allows learning hierarchical organization of word morphology as a collection of tree structured paradigms. The model is fully unsupervised and based on the hierarchical Dirichlet process. Tree hierarchies are learned along with the corresponding morphological paradigms simultaneously. Our model is evaluated on Morpho Challenge and shows competitive performance when compared to state-of-the-art unsupervised morphological segmentation systems. Although we apply this model for morphological segmentation, the model itself can also be used for hierarchical clustering of other types of data.*

## 1. Introduction

Unsupervised learning of morphology has been an important task because of the benefits it provides to many other natural language processing applications such as machine translation, information retrieval, question answering, and so forth. Morphological paradigms provide a natural way to capture the internal morphological structure of a group of morphologically related words. Following Goldsmith (2001) and Monson (2008), we use the term **paradigm** as consisting of a set of stems and a set of suffixes where each combination of a stem and a suffix leads to a valid word form, for example, {*walk*,*talk*,*order*,*yawn*}{*s*,*ed*,*ing*} generating the surface forms *walk+ed*, *walk+s*, *walk+ing*, *talk+ed*, *talk+s*, *talk+ing*, *order+s*, *order+ed*, *order+ing*, *yawn+ed*, *yawn+s*, *yawn+ing*. A sample paradigm is given in Figure 1.

Recently, we introduced a probabilistic hierarchical clustering model for learning hierarchical morphological paradigms (Can and Manandhar 2012). Each node in the hierarchical tree corresponds to a morphological paradigm and each leaf node consists of a word. A single tree is learned, where different branches on the hierarchical tree

Submission received: 29 June 2016; revised version received: 30 July 2017; accepted for publication: 1 March 2018.

doi:10.1162/COLI_a_00318

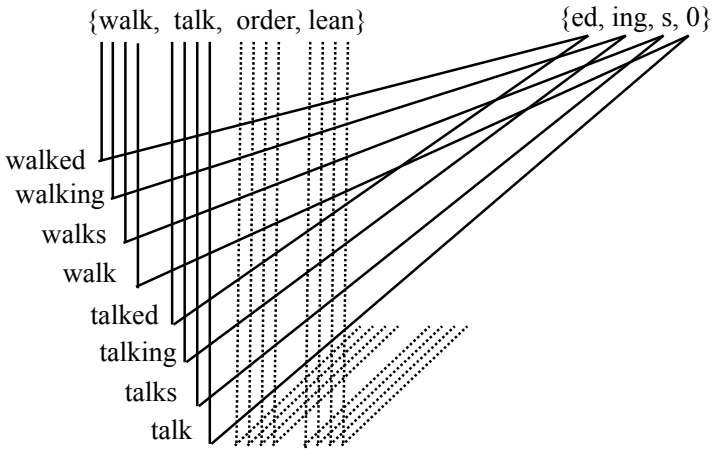



**Figure 1**
An example paradigm.



correspond to different morphological forms. Well-defined paradigms in the lower levels of trees were learned. However, merging of the paradigms at the upper levels led to undersegmentation in that model. This problem led us to search for ways to learn multiple trees. In our current approach, we learn a forest of paradigms spread over several hierarchical trees. Our evaluation on Morpho Challenge data sets provides better results when compared to the previous method (Can and Manandhar 2012). Our results are comparable to current state-of-the-art results while having the additional benefit of inferring the hierarchical structure of morphemes for which no comparable systems exist.

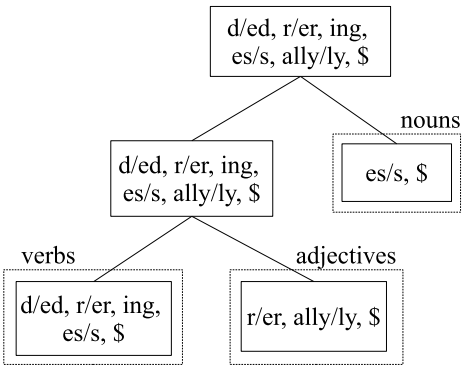
The article is organized as follows: Section 2 introduces the related work in the field. Section 3 describes the probabilistic hierarchical clustering model with its mathematical model definition and how it is applied for morphological segmentation; the same section explains the inference and the morphological segmentation. Section 4 presents the experimental setting and the obtained evaluation scores from each experiment, and Section 5 concludes and addresses the potential future work following the model presented in this article.

## 2. Related Work

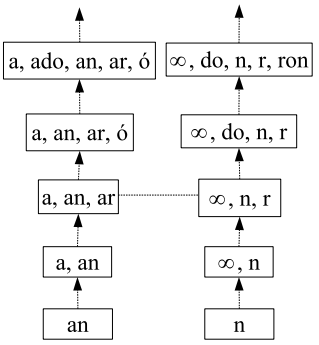
There have been many unsupervised approaches to morphology learning that focus solely on segmentation (Creutz and Lagus 2005a, 2007; Snyder and Barzilay 2008; Poon, Cherry, and Toutanova 2009; Narasimhan, Barzilay, and Jaakkola 2015). Others, such as Monson et al. (2008), Can and Manandhar (2010), Chan (2006), and Dreyer and Eisner (2011), learn morphological paradigms that permit additional generalization.

A popular paradigmatic model is Linguistica (Goldsmith 2001), which uses the Minimum Description Length principle to minimize the description length of a corpus based on paradigm-like structures called **signatures**. A signature consists of a list of suffixes that are seen with a particular stem—for example, *order*-{*ed, ing, s*} denotes a signature for the stem *order*.

Snover, Jarosz, and Brent (2002) propose a generative probabilistic model that defines a probability distribution over different segmentations of the lexicon into paradigms. Paradigms are learned with a directed search algorithm that examines




(a) Morphological paradigms in the recursion tree of Chan (2006).




(b) Paradigms in Spanish on a lattice structure in Paramor (Monson et al. 2008).

**Figure 2**
Examples of hierarchical morphological paradigms.



subsets of the lexicon, ranks them, and incrementally combines them in order to find the best segmentation of the lexicon. The proposed model addresses both inflectional and derivational morphology in a language independent framework. However, their model does not allow multiple suffixation (e.g., having multiple suffixes added to a single stem) whereas Linguistica allows this.

Monson et al. (2008) induce morphological paradigms in a deterministic framework named ParaMor. Their search algorithm begins with a set of candidate suffixes and collects candidate stems that attach to the suffixes (see Figure 2(b)). The algorithm gradually develops paradigms following search paths in a lattice-like structure. Probabilistic ParaMor, involving a statistical natural language tagger to mimic ParaMor, was introduced in Morpho Challenge 2009 (Kurimo et al. 2009). The system outperforms other unsupervised morphological segmentation systems that competed in Morpho Challenge 2009 (Kurimo et al. 2009) for the languages Finnish, Turkish, and German.

Can and Manandhar (2010) exploit syntactic categories to capture morphological paradigms. In a deterministic scheme, morphological paradigms are learned by pairing syntactic categories and identifying common suffixes between them. The paradigms compete to acquire more word pairs.

Chan (2006) describes a supervised procedure to find morphological paradigms within a hierarchical structure by applying latent Dirichlet allocation. Each paradigm is placed in a node on the tree (see Figure 2(a)). The results show that each paradigm corresponds to a part-of-speech such as adjective, noun, verb, or adverb. However, as the method is supervised, the true suffixes and segmentations are known in advance. Learning hierarchical paradigms helps not only in learning morphological segmentation, but also in learning syntactic categories. This linguistic motivation led us toward learning the hierarchical organization of morphological paradigms.

Dreyer and Eisner (2011) propose a Dirichlet process mixture model to learn paradigms. The model uses 50–100 seed paradigms to infer the remaining paradigms, which makes it semi-supervised. The model is similar to ours in the sense that it also uses Dirichlet processes (DPs); however the model does not learn hierarchies between the paradigms.

Luo, Narasimhan, and Barzilay (2017) learn morphological families that share the same stem, such as *faithful*, *faithfully*, *unfaithful*, *faithless*, and so on, that are all derived from *faith*. Those morphological families are learned as a graph and called **morphological forests**, which deviates from the meaning of the term *forest* we refer in this article. Although learning morphological families has been studied as a graph learning problem in Luo, Narasimhan, and Barzilay (2017), in this work, we learn paradigms that generalize morphological families within a collection of hierarchical structures.

Narasimhan, Barzilay, and Jaakkola (2015) model the word formation with morphological chains in terms of parent-child relations. For example, *play* and *playful* have a parent–child relationship as a result of adding the morpheme *ful* at the end of *play*. These relations are modeled by using log-linear models in order to predict the parent relations. Semantic features as given by word2vec (Mikolov et al. 2013) are used in their model in addition to orthographic features for the prediction of parent–child relations. Narasimhan, Barzilay, and Jaakkola use contrastive estimation and generate corrupted examples as pseudo negative examples within their approach.

Our model is an extension of our previous hierarchical clustering algorithm (Can and Manandhar 2012). In that algorithm, a single tree is learned that corresponds to a hierarchical organization of morphological paradigms. The parent nodes merge the paradigms from the child nodes. But such merging of paradigms into a single structure causes unrelated paradigms to be merged resulting in lower segmentation accuracy. The current model addresses this issue by learning a forest of tree structures. Within each tree structure the parent nodes merge the paradigms from the child nodes. Multiple trees ensure that paradigms that should not be merged are kept separated. Additionally, in single tree hierarchical clustering, a manually defined context free grammar was employed to generate the segmentation of a word. In the current model, we predict the segmentation of a word without using any manually defined grammar rules.



## 3. Probabilistic Hierarchical Clustering

Chan (2006) showed that learning the hierarchy between morphological paradigms can help reveal latent relations in data. In the latent class model of Chan (2006), morphological paradigms in a tree structure can be linked to syntactic categories (i.e., part-of-speech tags). An example output of the model is given in Figure 2(a). Furthermore, tokens can be assigned to allomorphs or gender/conjugational variants in each paradigm.

Monson et al. (2008) showed that learning paradigms within a hierarchical model gives a strong performance on morphological segmentation. In their model, each paradigm is part of another paradigm, implemented within a lattice structure (see Figure 2(b)).

Motivated by these works, we aim to learn paradigms within a hierarchical tree structure. We propose a novel hierarchical clustering model that deviates from the current hierarchical clustering models in two aspects:

1. It is a generative model based on a hierarchical Dirichlet process (HDP) that simultaneously infers the hierarchical structure and morphological segmentation.
2. Our model infers multiple trees (i.e., a forest of trees) instead of a single tree.

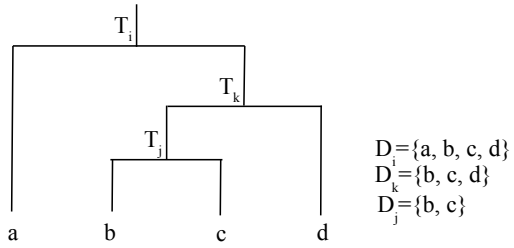



**Figure 3**
A portion of a tree rooted at $i$ with child nodes $k$, $j$ with corresponding data points they generate. $i$, $j$, $k$ refer to the tree nodes; $T_i$, $T_j$, $T_k$ refer to corresponding trees; and $D_i$, $D_j$, $D_k$ refer to data contained in trees.

Although not covered in this work, additional work can be aimed at discovering other types of latent information such as part-of-speech and allomorphs.

### 3.1 Model Overview

Let $D = \{x^1, x^2, \ldots, x^N\}$ denote the input data where each $x^j$ is a word. Let $D_i \subseteq D$ denote the subset of the data generated by tree rooted at $i$, then (see Figure 3):

$$D = D_1 \cup D_2 \cup \cdots \cup D_k \tag{1}$$

where $D_i = \{x_i^1, x_i^2, \ldots, x_i^{N_i}\}$.

The marginal probability of the data items in a given node in a tree $i$ with parameters $\theta$ and hyperparameters $\beta$ is given by:

$$p(D_i) = \int p(D_i|\theta)p(\theta|\beta)d\theta \tag{2}$$

Given tree $T_i$ the data likelihood is given recursively by:

$$p(D_i|T_i) = \begin{cases} \frac{1}{Z}p(D_i)p(D_l|T_l)p(D_r|T_r) & \text{if } i \text{ is an internal node with child nodes } l \text{ and } r, \text{ and } Z \text{ is the partition function} \\ p(D_i) & \text{if } i \text{ is a leaf node} \end{cases} \tag{3}$$

The term $\frac{1}{Z}(D_i)p(D_l|T_l)p(D_r|T_r)$ corresponds to a **product of experts** model (Hinton 2002) comprising *two* competing hypotheses.[1] Hypothesis $H_1$ is given by $p(D_i)$ and assigns all the data points in $D_i$ into a single cluster. Hypothesis $H_2$ is given by

1 The review version of this paper and our previous work (Can and Manandhar 2012) missed this connection to the product of experts model and did not include the 1/Z term. We thank our reviewers for spotting this error.

$p(D_l|T_l)p(D_r|T_r)$ and recursively splits the data in $D_i$ into two partitions (i.e., subtrees) $D_l$ and $D_r$. The factor $Z$ is the partition function or the normalizing constant given by:

$$Z = \sum_{D_i, D_l, D_r : D_i = D_l \cup D_r} p(D_i)p(D_l|T_l)p(D_r|T_r) \tag{4}$$

The recursive partitioning scheme we use is similar to that of Heller and Ghahramani (2005). The product of experts scheme used in this paper contrasts with the more conventional sum of experts mixture model of Heller and Ghahramani (2005) that would have resulted in the mixture $\pi\ p(D_i) + (1 - \pi)\ p(D_l|T_l)p(D_r|T_r)$, where $\pi$ denotes the mixing proportion.

Finally, given the collection of trees $T = \{T_1, T_2, \ldots, T_n\}$, the total likelihood of data in all trees is defined as follows:

$$p(D|T) = \prod_{i}^{|T|} p(D_i|T_i) \tag{5}$$

Trees are generated from a DP. Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of trees. $T_i$ is sampled from a DP as follows:

$$F \sim DP(\alpha, U) \tag{6}$$

$$T_i \sim F \tag{7}$$

where $\alpha$ denotes the concentration parameter of the DP. $U$ is a uniform base distribution that assigns equal probability to each tree. We integrate out $F$, which is a distribution over trees, instead of estimating it. Hence, the conditional probability of sampling an existing tree is computed as follows:

$$p(T_k|T, \alpha, U) = \frac{N_k}{N + \alpha} \tag{8}$$

where $N_k$ denotes the number of words in $T_k$ and $N$ denotes the total number of words in the model. A new tree is generated with the following:

$$p(T_{|T|+1}|T, \alpha, U) = \frac{\alpha/k}{N + \alpha} \tag{9}$$

### 3.2 Modeling Morphology with Probabilistic Hierarchical Clustering

In our model, data points are the words and each tree node corresponds to a morphological paradigm (see Figure 4). Each word is part of all the paradigms on the path from the leaf node having that word until the root node. The word can share either its stem or suffix with other words in the same paradigm. Hence, a considerable number of words can be generated through this approach that may not be seen in the corpus. Our model will prefer words that share stems or suffixes to be close to each other within the tree.

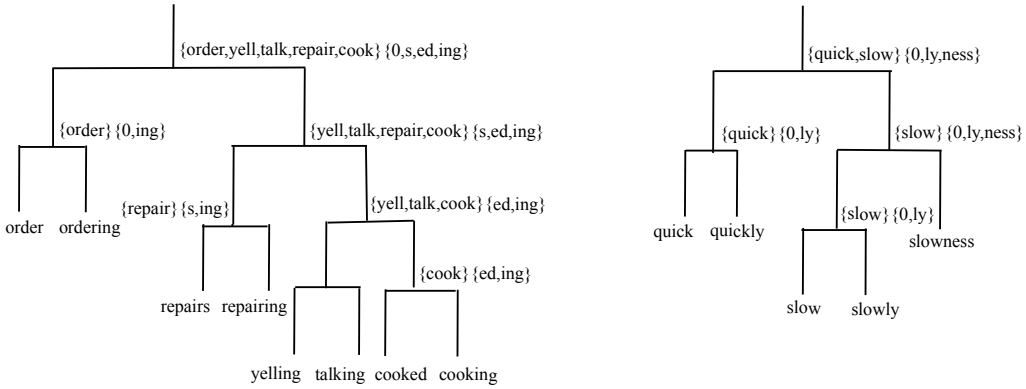



**Figure 4**
A sample hierarchical tree structure that illustrates the clusters in each node (i.e., paradigms). Each node corresponds to a cluster (i.e., morphological paradigm) and the leaf nodes correspond to input data. The figure shows the ideal forest of trees that one expects given the input data.

The plate diagram of the generative model is given in Figure 5(a). Given a child node $i$, we define a Dirichlet process to generate stems (denoted by $s_i^1, \ldots, s_i^{N_i}$) and a separate Dirichlet process to generate suffixes (denoted by $m_i^1, \ldots, m_i^{N_i}$):

$$G_i^s \sim DP(\beta_s, P^s) \tag{10}$$

$$s_i^j \sim G_i^s \tag{11}$$

$$G_i^m \sim DP(\beta_m, P^m) \tag{12}$$

$$m_i^j \sim G_i^m \tag{13}$$

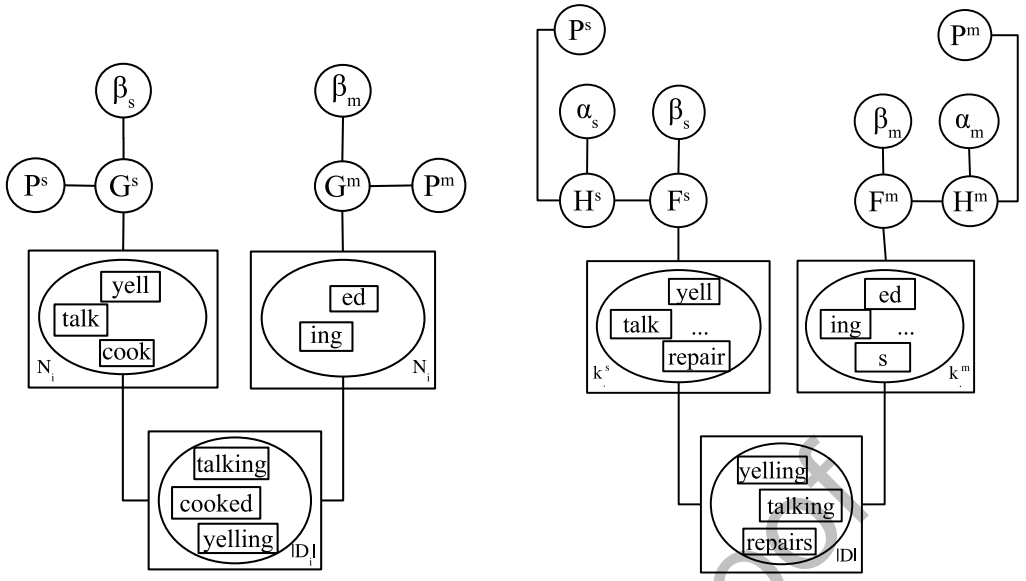
$$P^s(s_i^j) = \prod_{c \in s_i^j} p(c) \tag{14}$$

$$P^m(m_i^j) = \prod_{c \in m_i^j} p(c) \tag{15}$$

where $DP(\beta_s, P^s)$ is a Dirichlet process that generates stems, $\beta_s$ denotes the concentration parameter, and $P^s$ is the base distribution. $G_i^s$ is a distribution over the stems $s_i^j$ in node $i$. Correspondingly, $DP(\beta_m, P^m)$ is a Dirichlet process that generates suffixes with analogous parameters. $G_i^m$ is a distribution over the suffixes $m_i^j$ in node $i$. For smaller values of the concentration parameter, it is less likely to generate new types. Thus, sparse multinomials can be generated by the Dirichlet process yielding a skewed distribution. We set $\beta_s < 1$ and $\beta_m < 1$ in order to generate a small number of stem and suffix types. $s_i^j$ and $m_i^j$ are the $j$th stem and suffix instance in the $i$th node, respectively.

The base distributions for stems and suffixes are given by Equation (14) and Equation (15). Here, $c$ denotes a single letter or character. We assume that letters are distributed uniformly (Creutz and Lagus 2005b), where $p(c) = 1/A$ for an alphabet having $A$ letters. Our model will favor shorter morphemes because they have less factors in the joint probability given by Equations (14) and (15).

For the root nodes we use HDPs that share global DPs (denoted by $H^s$ for stems and $H^m$ for suffixes). These introduce dependencies between stems/suffixes in distinct




(a) The plate diagram of the DP model (b) The plate diagram of the HDP model

**Figure 5**
The DP model for the child nodes (on the left) illustrates the generation of words *talking*, *cooked*, *yelling*. Each child node maintains its own DP that is independent of DPs from other nodes. The HDP model for the root nodes (on the right) illustrates the generation of words *yelling*, *talking*, *repairs*. In contrast to the DPs in the child nodes, in the HDP model, the stems/suffixes are shared across all root HDPs.



trees. The model will favor stems and suffixes that are already generated in one of the trees. The HDP for a root node $i$ is defined as follows:

$$F_i^s \sim DP(\beta_s, H^s) \tag{16}$$
$$H^s \sim DP(\alpha_s, P^s) \tag{17}$$
$$s_i^j \sim F_i^s \tag{18}$$
$$\psi_i^z \sim H^s \tag{19}$$
$$F_i^m \sim DP(\beta_m, H^m) \tag{20}$$
$$H^m \sim DP(\alpha_m, P^m) \tag{21}$$
$$m_i^j \sim F_i^m \tag{22}$$
$$\phi_i^z \sim H^m \tag{23}$$

where the base distributions $H^s$ and $H^m$ are drawn from the global DPs $DP(\alpha_s, P^s)$ and $DP(\alpha_m, P^m)$. Here, $\psi_i^z$ denotes the stem type $z$ in node $i$, and $\phi_i^z$ denotes the suffix type $z$ in node $i$, which are drawn from $H_s$ and $H_m$ (i.e., the global DPs), respectively. The plate diagram of the HDP model is given in Figure 5(b).

From the Chinese restaurant process (CRP) metaphor perspective, the global DPs generate the global sets of dishes (i.e., stems and suffixes) that constitute the menu for all trees in the model. At each tree node, there are two restaurants: one for stems and one for suffixes. At each table, a different type of dish (i.e., stem or suffix type) is served

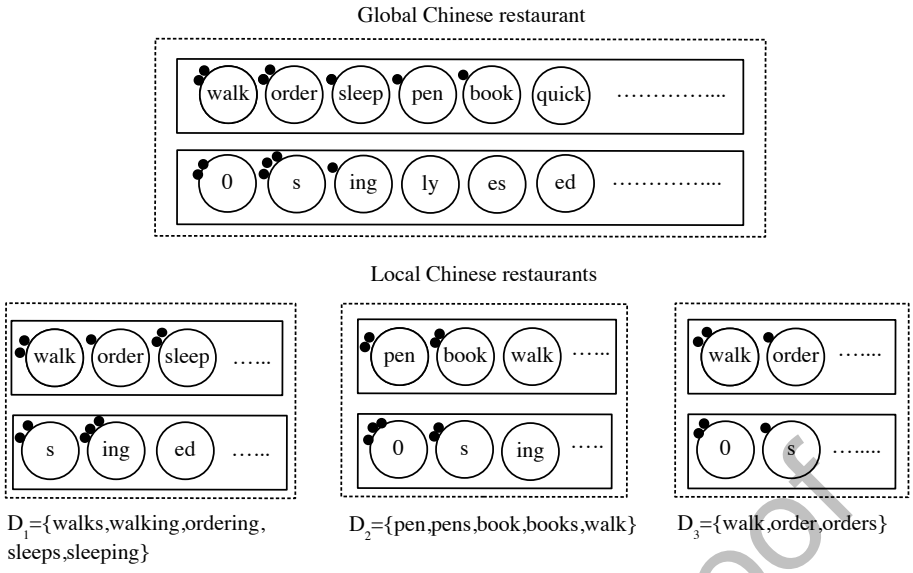



**Figure 6**
A depiction of the Chinese restaurant franchise (i.e., global vs. local CRPs). $S_1 = \{walk, order, sleep, etc.\}$, $M_1 = \{s, ing\}$, $S_2 = \{pen, book\}$, $M_2 = \{0, s\}$, $S_3 = \{walk, order\}$, $M_3 = \{0, s\}$, where 0 denotes an empty suffix. For each stem type in the distinct trees, a customer is inserted in the global restaurant. For example, there are two stem customers that are being served the stem type *walk* because *walk* exists in two different trees.

and for each stem/suffix type there exists only one table in each node (i.e., restaurant). Customers are the stem or suffix tokens. Whenever a new customer, $s_i^j$ or $m_i^j$, enters a restaurant, if the table, $\psi_i^z$ or $\phi_i^z$, serving that dish already exists, the new customer sits at that table. Otherwise, a new table is generated in the restaurant. A change in one of the restaurants in the leaf nodes leads to the update in each restaurant all the way to the root node. If the dish is not available in the root node, a new table is created for that root node and a global customer is also added to the global restaurant. If no global table exists for that dish, a new global table serving the dish is also created. This can be seen as a Chinese restaurant franchise where each node is a restaurant itself (see Figure 6).

In order to calculate the joint likelihood of the model, we need to consider both trees and global stem/suffix sets (i.e., local and global restaurants). The model is exchangeable because it is a CRP—which means that the order the words are segmented does not alter the joint probability. The joint likelihood of the entire model for a given collection of trees $T = \{T_1, T_2, \ldots, T_n\}$ is computed as follows:

$$p(D|T) = \prod_{i}^{|T|} p(D_i|T_i) \tag{24}$$

$$= \prod_{i}^{|T|} p(S_i|T_i)p(M_i|T_i) \tag{25}$$

$$= p(S_{root})p(M_{root}) \prod_{\substack{i=1 \\ i \neq root}}^{|T|} p(S_i|T_i)p(M_i|T_i) \tag{26}$$

where $p(S_i|T_i)$ and $p(M_i|T_i)$ are computed recursively:

$$p(S_i|T_i) = \begin{cases} \frac{1}{Z} p(S_i)p(S_l|T_l)p(S_r|T_r) & \text{if } i \text{ is an internal node with child nodes } l \text{ and } r \\ p(S_i) & \text{if } i \text{ is a leaf node} \end{cases} \tag{27}$$

where $Z$ is the normalization constant. The same also applies for $M_i$.

Following the CRP, the joint probability of the stems in each root node $T_i$, $S_{root_i} = \{s_i^1, s_i^2, \ldots, s_i^{N_i}\}$, is:

$$p(S_{root}) = \prod_i^{|T|} p(s_i^1, s_i^2, \ldots, s_i^{N_i}) \tag{28}$$

$$= \prod_i^{|T|} p(s_i^1)p(s_i^2|s_i^1)\ldots p(s_i^{N_i}|s_i^1, \ldots, s_i^{N_i-1}) \tag{29}$$

$$= \left( \prod_i^{|T|} \left( \frac{\Gamma(\beta_s)}{\Gamma(N_i + \beta_s)} \beta_s^{L_i^s - 1} \prod_{j=1}^{L_i^s} (n_{ij}^s - 1)! \right) \right) \tag{30}$$

$$\left( \frac{\Gamma(\alpha_s)}{\Gamma(\sum_{s_j \in K^s} k_j^s + \alpha_s)} \alpha_s^{K^s - 1} \prod_{j=1}^{K^s} (k_j^s - 1)! P^s(s_j) \right)$$

where the first line in Equation (30) corresponds to the stem CRPs in the root nodes of the trees (see Equation (2.22) in Can [2011] and Equation (A.1) in Appendix A). The second line in Equation (30) corresponds to the global CRP of stems. The second factor in the first line of Equation (30) corresponds to the case where $L_i^s$ stem types are generated the first time, and the third factor in the first line corresponds to the case, where for each of the $L_i^s$ stem types at node $i$, there are $n_{ij}^s$ stem tokens of type $j$. The first factor accounts for all denominators from both cases. Similarly, the second and the fourth factor in the second line of Equation (30) corresponds to the case where $K^s$ stem types are generated globally (i.e., stem tables in the global restaurant), the third factor corresponds to the case where, for each of the $K^s$ stem types, there are $k_j^s$ stems of type $j$ in distinct trees. A sample hierarchical structure is given in Figure 7.

The joint probability of stems in a child node $T_i$, $S_i = \{s_i^1, s_i^2, \ldots, s_i^{N_i}\}$, which belong to stem tables $\{\psi_i^1, \psi_i^2, \ldots\}$ that index the items on the global menu is reduced to:

$$p(S_i) = \frac{\Gamma(\beta_s)}{\Gamma(N_i + \beta_s)} \beta_s^{L_i^s - 1} \prod_{j=1}^{L_i^s} \left( (n_{ij}^s - 1)! P^s(\psi_i^j) \right) \tag{31}$$

Whenever a new stem is added to a node $i$ (i.e., new customer enters one of the local restaurants), the conditional probability of the new stem $s_i^j$ that belongs to type $z$

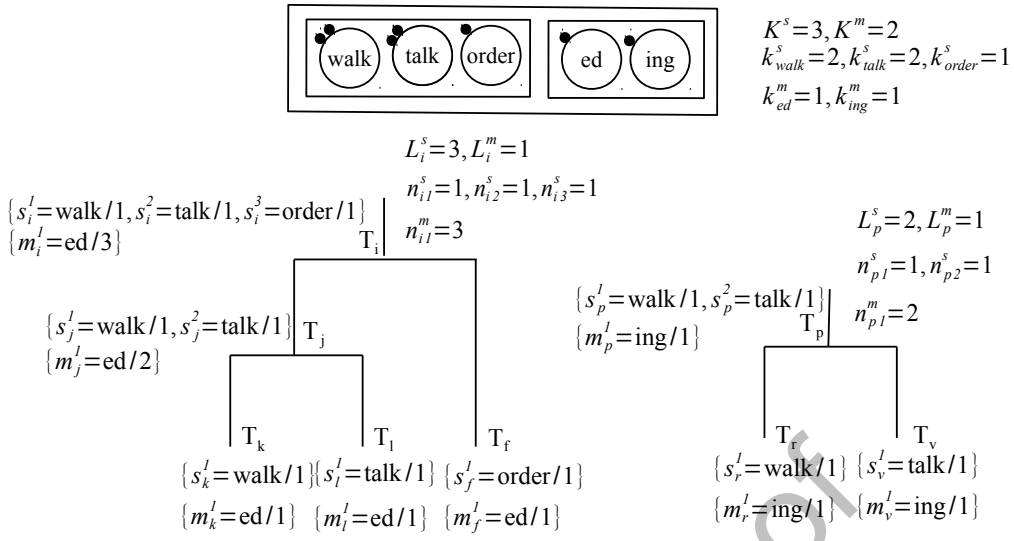



**Figure 7**
A sample hierarchical structure that contains $D = \{walk + ed, talk + ed, order + ed, walk + ing, talk + ing\}$ and the corresponding global tables. Here $s^1_k = walk/1$ denotes the first stem *walk* in node $k$ with frequency 1.

(i.e., customer $s^j_i$ sitting at table $\psi^z_i$) is computed as follows (see Teh 2010 and Teh et al. 2006):

$$p(s^j_i = z | S_i^{-s^j_i}, \beta_s, P^s) = \begin{cases} \dfrac{n^{-s^j_i}_{iz}}{N_i - 1 + \beta_s} & \text{if } \psi^z_i \in \Psi_i \\ \dfrac{\beta_s P^s(s^j_i)}{N_i - 1 + \beta_s} & \text{otherwise (for an internal node)} \\ \dfrac{\beta_s H^s(\psi^z_i | \alpha_s, P^s)}{N_i - 1 + \beta_s} & \text{otherwise (for a root node)} \end{cases} \tag{32}$$

where $\Psi_i$ denotes the table indicators in node $i$, $n^{-s^j_i}_{iz}$ denotes the number of stem tokens that belong to type $z$ in node $i$ when the last instance $s^j_i$ is excluded.

If the stem (customer) does not exist in the root node (i.e., chooses a non-existing dish type in the root node), the new stem's probability is calculated as follows:

$$H^s(\psi^z_i | \alpha_s, P^s) = \begin{cases} \dfrac{k^s_z}{\sum_{s_j \in K^s} k^s_j + \alpha_s} & \text{if } j \in K^s \\ \dfrac{\alpha_s P^s(s_z)}{\sum_{s_j \in K^s} k^s_j + \alpha_s} & \text{otherwise} \end{cases} \tag{33}$$

If the stem type (dish) exists in the global clusters (i.e., global menu), it is chosen with the probability proportional to the number of trees that contain the stem type (i.e., number of global stem customers that are served the dish). Otherwise, the new stem (dish type) is chosen based on the base distribution $P^s$.

Analogously to Equations (30)–(33) that apply to stems, the corresponding equations for suffixes are given by Equations (34)–(37).

$$p(M_{root}) = \left(\prod_{i}^{|T|}\left(\frac{\Gamma(\beta_m)}{\Gamma(N_i+\beta_m)}\beta_m^{L_i^m-1}\prod_{j=1}^{L_i^m}(n_{ij}^m-1)!\right)\right) \left(\frac{\Gamma(\alpha_m)}{\Gamma(\sum_{m_j\in K^m}k_j^m+\alpha_m)}\alpha_m^{K^m-1}\prod_{j=1}^{K^m}(k_j^m-1)!P^m(m_j)\right) \tag{34}$$

$$p(M_i) = \frac{\Gamma(\beta_m)}{\Gamma(N_i+\beta_m)}\beta_m^{L_i^m-1}\prod_{j=1}^{L_i^m}\left((n_{ij}^m-1)!P^m(\phi_i^j)\right) \tag{35}$$

$$p(m_i^j = z|M_i^{-m_i^j},\beta_m,P^m) = \begin{cases} \frac{n_{iz}^{-m_i^j}}{N_i-1+\beta_m} & \text{if } \phi_i^z\in\Phi_i \\ \frac{\beta_m P^m(m_i^j)}{N_i-1+\beta_m} & \text{otherwise (for an internal node)} \\ \frac{\beta_m H^m(\phi_i^z|\alpha_m,P^m)}{N_i-1+\beta_m} & \text{otherwise (for a root node)} \end{cases} \tag{36}$$

$$H^m(\phi_i^z|\alpha_m,P^m) = \begin{cases} \frac{k_z^m}{\sum_{m_j\in K^m}k_j^m+\alpha_m} & \text{if } j\in K^m \\ \frac{\alpha_m P^m(m_z)}{\sum_{m_j\in K^m}k_j^m+\alpha_m} & \text{otherwise} \end{cases} \tag{37}$$

where $M_i = \{m_i^1, m_i^1, \ldots, m_i^{N_i}\}$ is the set of suffixes in node $i$ belonging to global suffix types $\{\phi_i^1, \phi_i^2, \ldots\}$; $N_i^m$ is the number of local suffix types; $n_{ij}^m$ is the number of suffix tokens of type $j$ in node $i$; $K^m$ is the total number of suffix types; and $k_j^m$ is the number of trees that contain suffixes of type $j$.

### 3.3 Metropolis-Hastings Sampling for Inferring Trees

Trees are learned along with the segmentations of words via inference. Learning trees involves two steps: 1) constructing initial trees; 2) Metropolis-Hastings sampling for inferring trees.

*3.3.1 Constructing Initial Trees.* Initially, all words are split at random points with uniform probability. We use an approximation to construct the intial trees. Instead of computing the full likelihood of the model, for each tree, we only compute the likelihood of a single DP and assume that all words belong to this DP. The conditional probability of inserting word $w^j = s + m$ in tree $T_k$ is given by:

$$\begin{aligned} p(T_k, w^j = s+m|D,T,\alpha,\beta_s,\beta_m,P^s,P^m,U) \\ &= p(T_k|T,\alpha,U)p(w^j = s+m|D_k) \\ &= p(T_k|T,\alpha,U)p(s|S_k,\beta_s,P^s)p(m|M_k,\beta_m,P^m) \end{aligned} \tag{38}$$

We use Equation (8) and (9) for computing the conditional probability $p(T_k|T, \alpha, U)$ of choosing a particular tree. Once a tree is chosen, a branch to insert is selected at random. The algorithm for constructing the initial trees is given in Algorithm 1.

*3.3.2 Metropolis-Hastings Sampling.* Once the initial trees are constructed, the hierarchical structures yielding a global maximum likelihood are inferred by using the Metropolis Hastings algorithm (Hastings 1970). The inference is performed by iteratively removing a word from a leaf node from a tree and subsequently sampling a new tree, a position within the tree, and a new segmentation (see Algorithm 2). Trees are sampled with the conditional probability given in Equations (8) and (9). Hence, trees with more words attract more words, and new trees are created proportional to the hyperparameter $\alpha$.

Once a tree is sampled, we draw a new position and a new segmentation. The word is inserted at the sampled position with the sampled segmentation. The new model is either accepted or rejected with the Metropolis-Hastings accept-reject criteria. We also use a simulated annealing cooling schedule by assigning an initial temperature $\gamma$ to the system and decreasing the temperature at each iteration. The accept probability we use is given by:

$$P_{Acc} = \left( \frac{p_{next}(D|T)}{p_{cur}(D|T)} \right)^{\frac{1}{\gamma}} \tag{39}$$

where $p_{next}(D|T)$ denotes the likelihood of the data under the altered model and $p_{cur}(D|T)$ denotes the likelihood of data under the current model before sampling. The normalization constant cancels out because it is the same for the numerator and the denominator. If $p_{next}(D|T)^{\frac{1}{\gamma}} > p_{cur}(D|T)^{\frac{1}{\gamma}}$, then the new sample is accepted: otherwise,



**Algorithm 1** Construction of the initial trees.

**input:** data $D := \{w^1 = s^1 + m^1, \ldots, w^N = s^N + m^N\}$
**initialize:** The trees in the model: $T := \emptyset$
**For** $j = 1 \ldots N$
Choose $w^j$ from the corpus.
Sample, $T_k$ as a new empty tree or existing tree from $T$, and, a split point $w^j = s + m$ from the joint distribution (see Equation 38):
$p(T_k, w^j = s + m|D, T, \alpha, \beta_s, \beta_m, P^s, P^m, U)$
**if** $T_k \in$ **T then**
    Draw a node $s$ from $T_k$ with uniform probability
    Assign $T_k$ as a sibling of $T_s$: $T_s := T_s + T_k$ (i.e., $T_s$ is new tree with $T_k$ and old $T_s$ as children)
**else**
    $T_k$ is an empty tree. Add $T_k$ into $T$: $T := T \cup T_k$
    $D_k := \{w^j = s + m\}$ (i.e., word $w^j = s + m$ is assigned into $T_k$)
**end if**
Remove $w^j$ from the corpus
**End For**
**output: T**

**Algorithm 2** Learning the trees with Metropolis-Hastings algorithm

**input:** data $D := \{w^1 = s^1 + m^1, \ldots, w^N = s^N + m^N\}$, initial trees $T$, initial temperature $\gamma$, the target temperature $\kappa$, temperature decrement $\eta$
**initialize:** $p_{cur}(D|T) := p(D|T)$
**while** $\gamma > \kappa$ **do**
  Choose the leaf node $j$ from **all** trees with uniform probability
  Let $D_j := \{w^j = s^{old} + m^{old}\}$
  Draw a split point $w^j = s^{new} + m^{new}$ with uniform probability
  Draw a tree $T_k$ with probability $p(T_k|T, \alpha, U)$ (see Equations 8 and 9)
  **if** $T_k \in T$ **then**
    Draw a sibling node $s$ from $T_k$ with uniform probability
    $T_s := T_s + T_k$ (see Figure 8)
  **else**
    Create a new tree $T_k$
    $T := T \cup T_k$
    $D_k := \{w^j = s^{new} + m^{new}\}$
  **end if**
  $rand \sim Normal(0, 1)$
  $p_{next} := p(D|T)$ (see Equation 24)
  **if** $p_{next}(D|T) >= p_{cur}(D|T)$ **or** $rand < \left(\frac{p_{next}(D|T)}{p_{cur}(D|T)}\right)^{\frac{1}{\gamma}}$ **then**
    Accept the new tree structure
    $p_{cur}(D|T) := p_{next}(D|T)$
  **end if**
  $\gamma := \gamma - \eta$
**end while**
**output: T**



the new model is still accepted with a probability $p_{Acc}$. The system is cooled in each iteration with decrements $\eta$. We refer to Section 4 for details of parameter settings.

$$\gamma \leftarrow \gamma - \eta \tag{40}$$

### 3.4 Morphological Segmentation

Once the model is learned, it can be used for the segmentation of novel words. We use only root nodes for the segmentation. Viterbi decoding is used in order to find the morphological segmentation of each word having the maximum probability:

$$\begin{aligned}\arg \max_{s^1,\cdots,s^a,m^1,\ldots,m^b} p(w^k = s^1, \cdots, s^a, m^1, \cdots, m^b | D, \alpha_s, \beta_s, H^s, P^s, \alpha_m, \beta_m, H^m, P^m) \\ = \prod_{j=1}^{a} \sum_{\substack{i=1 \\ i=root}}^{|T|} p(s_i^j | S_i, \alpha_s, \beta_s, H^s, P^s) \\ \prod_{j=1}^{b} \sum_{\substack{i=1 \\ i=root}}^{|T|} p(m_i^j | M_i, \alpha_m, \beta_m, H^m, P^m)\end{aligned} \tag{41}$$

where $w^k$ denotes the kth word to be segmented in the test set.

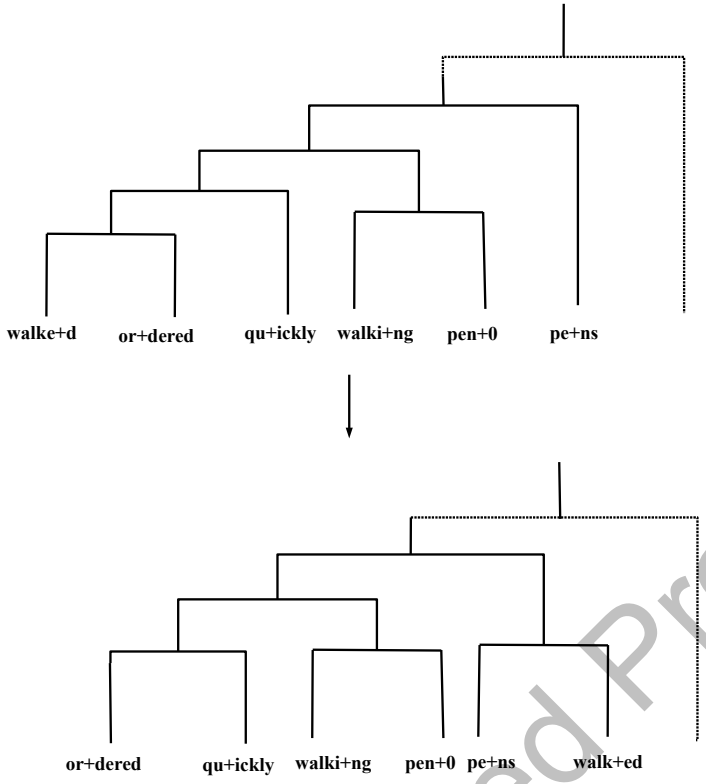



**Figure 8**
A sampling step in Metropolis-Hastings algorithm.



### 3.5 Example Paradigms

A sample of root paradigms learned by our model for English is given in Table 1. The model can find similar word forms (i.e., separat+ists, medal+ists, hygien+ists) that are grouped in the neighbor branches in the tree structure (see Figures 9, B.1, and B.2 for sample paradigms learned in English, Turkish, and German).

Paradigms are captured based on the similarity of either stems or suffixes. Having the same stem such as co-chair (co-chairman, co-chairmen) or trades (trades+man, trades+men) allows us to find segmentations such as co-chair+man vs. co-chair+men and trades+man vs. trades+men. Although we assume a stem+suffix segmentation, other types of segmentation, such as prefix+stem, are also covered. However, stem alterations and infixation are not covered in our model.

## 4. Evaluation

We used publicly available Morpho Challenge data sets for English, German, and Turkish for training. The English data set consists of 878,034 words, the German data set consists of 2,338,323 words, and the Turkish data set consists of 617,298 words. Although frequency of each word was available in the training set, we did not make use of this

**Table 1**
Example paradigms in English.

| |
|---|
| {final, ungod, frequent, pensive} {ly} |
| {kind, kind, kind} {est, er, 0} |
| {underrepresent, modul} {ation} |
| {plebe, hawai, muslim-croat} {ian} |
| {compuls, shredd, compuls, shredd, compuls} {ion, er, ively, ers, ory} |
| {zion, modern, modern, dynam} {ism, ists} |
| {reclaim, chas, pleas, fell} {ing} |
| {mov, engrav, stray, engrav, fantasiz, reischau, decilit, suspect} {ing, er, e} |
| {measur, measur, incontest, transport, unplay, reput} {e, able} |
| {housewar, decorat, entitl, cuss, decorat, entitl, materi, toss, flay, unconfirm linse, equipp} {es, ing, alise, ed} |
| {fair, norw, soon, smooth, narrow, sadd, steep, noisi, statesw, narrow} {est, ing} |
| {rest, wit, name, top, odor, bay, odor, sleep} {less, s} |
| {umpir, absorb, regard, embellish, freez, gnash} {ing} |
| {nutrition, manicur, separat, medal, hygien, nutrition, genetic, preservation} {0, ists} |



information. In other words, we use only the word types (not tokens) in training. We do not address the ambiguity of words in this work and leave this as future research.

In all experiments, the initial temperature of the system is set $\gamma = 2$ and it is reduced to $\gamma = 0.01$ with decrements $\eta = 0.0001$ (see Equation (39)). Figure 10 shows the time required for the log likelihoods of the trees of sizes 10K, 16K, and 22K to converge. We fixed $\alpha_s = \alpha_m = \beta_s = \beta_m = 0.01$ and $\alpha = 0.0005$ in all our experiments. The hyperparameters are set manually as a result of several experiments. These are the optimum values obtained from a number of experiments.[2]

Precision, recall, and F-score values against training set sizes are given in Figures 11 and 12 for English and Turkish, respectively.

### 4.1 Morpho Challenge Evaluation

Although we experimented with different sizes of training sets, we used a randomly chosen 600K words from the English and 200K words from the Turkish and German data sets for evaluation purposes.

Evaluation is performed according to the method proposed in Morpho Challenge (Kurimo et al. 2010a), which in turn is based on evaluation used by Creutz and Lagus (2007). The gold standard evaluation data set utilized within the Morpho Challenge is a hidden set that is *not* available publicly. This makes the Morpho Challenge evaluation different from other evaluations that provide test data. In this evaluation, word pairs

2 The source code of the model is accessible at: `https://github.com/burcu-can/TreeStructuredDP`.

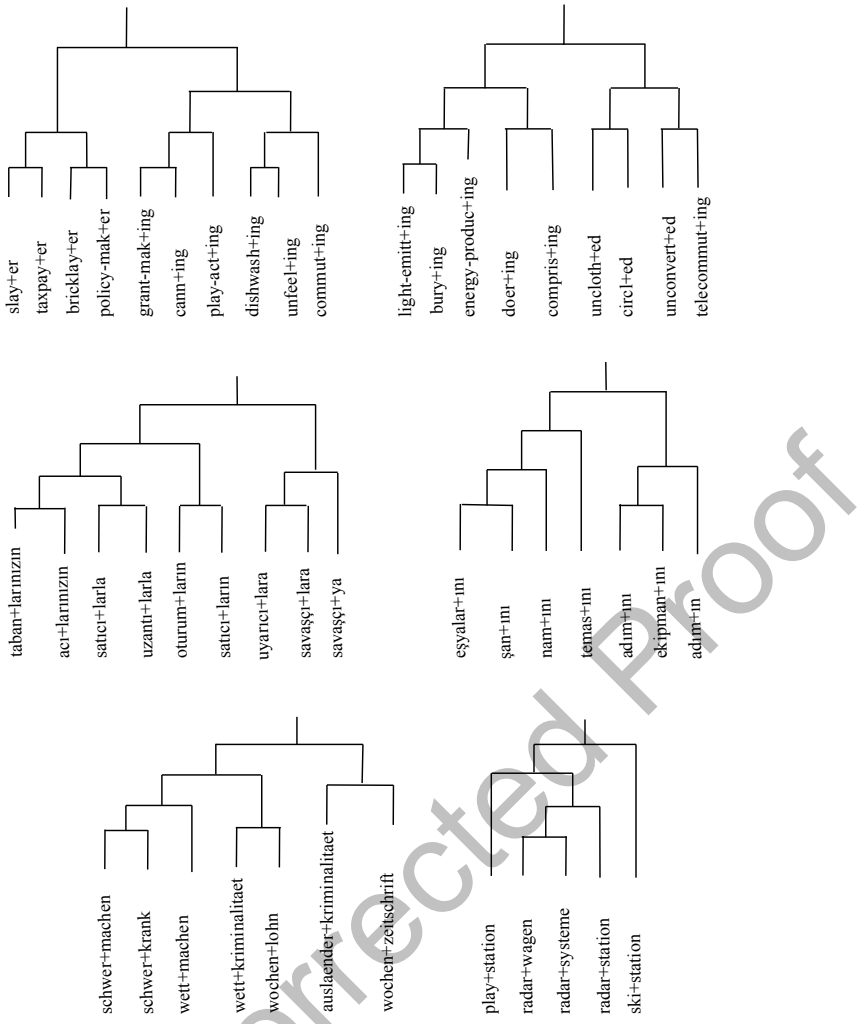



**Figure 9**
Sample hierarchies in English, Turkish, and German.



sharing at least one common morpheme are sampled. For precision, word pairs are sampled from the system results and checked against the gold standard segmentations. For recall, word pairs are sampled from the gold standard and checked against the system results. For each matching morpheme, 1 point is given. Precision and recall are calculated by normalizing the total obtained scores.

We compare our results with other unsupervised systems from Morpho Challenge 2010 (Kurimo et al. 2010b) for English, German, and Turkish. More specifically, we compare our model with all competing unsupervised systems: Morfessor Baseline (Creutz and Lagus 2002), Morfessor CATMAP (Creutz and Lagus 2005a), Base Inference (Lignos 2010), Iterative Compounding (Lignos 2010), Aggressive Compounding (Lignos 2010), and Nicolas, Farré, and Molinero (2010). Additionally, we compare our system with the Morpho Chain model of Narasimhan, Barzilay, and Jaakkola (2015) by re-training their model on exactly the same training sets as ours. All evaluation was carried out by the Morpho Challenge organizers based on the hidden gold data sets.

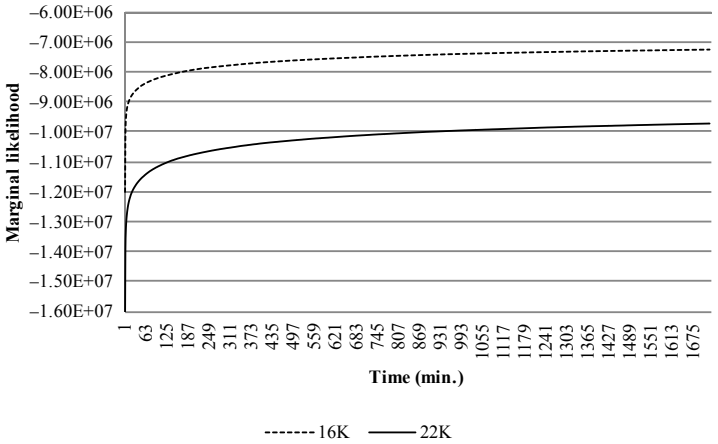



**Figure 10**
The likelihood convergence in time (in minutes) for data sets of size 16K and 22K.

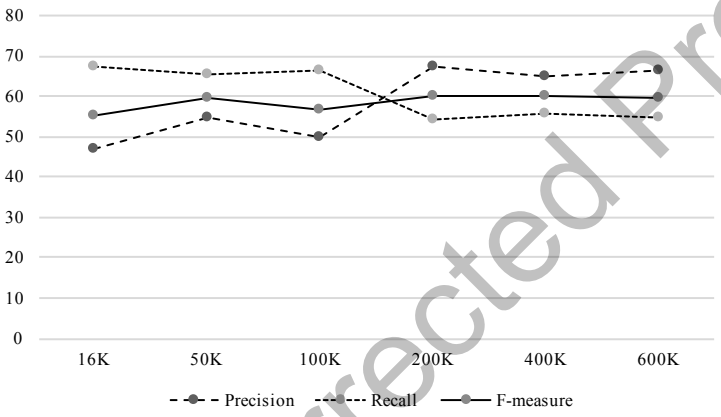



**Figure 11**
English results for different size of data.



English results are given in Table 2. For English, the Base Inference algorithm of Lignos (2010) obtained the highest F-measure in Morpho Challenge 2010 among other competing unsupervised systems. Our model is ranked fourth among all unsupervised systems with a F-measure 60.27%.

German results are given in Table 3. Our model outperforms other unsupervised models in Morpho Challenge 2010 with a F-measure 50.71%.

Turkish results are given in Table 4. Again, our model outperforms other unsupervised participants, achieving a F-measure 56.41%.

Our model also outperforms the model from Can and Manandhar (2012) (which we refer to as Single Tree Probabilistic Clustering), although the results are not directly comparable because the largest data set we were able to train that model on was 22K words due to the training time required. The full training set provided by Morpho Challenge was not used in Can and Manandhar (2012). Our current approach is more efficient as the training cost is divided across multiple tree structures with each tree being shallower compared with our previous model.

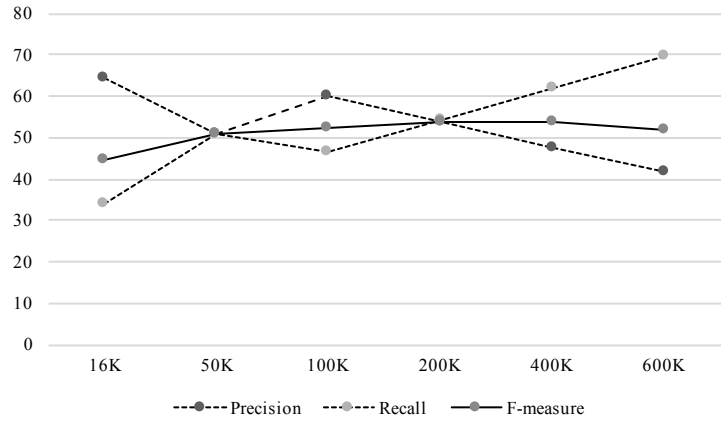



**Figure 12**
Turkish results for different size of data.

**Table 2**
Morpho Challenge 2010 experimental results for English.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 55.60 | **65.80** | 60.27 |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | 55.60 | 57.58 | 57.33 |
| Base Inference (Lignos 2010) | 80.77 | 53.76 | **64.55** |
| Iterative Compounding (Lignos 2010) | 80.27 | 52.76 | 63.67 |
| Aggressive Compounding (Lignos 2010) | 71.45 | 52.31 | 60.40 |
| Nicolas (Nicolas, Farré, and Molinero 2010) | 67.83 | 53.43 | 59.78 |
| Morfessor Baseline (Creutz and Lagus 2002) | 81.39 | 41.70 | 55.14 |
| Morpho Chain (Narasimhan, Barzilay, and Jaakkola 2015) | 74.87 | 39.01 | 50.42 |
| Morfessor CatMAP (Creutz and Lagus 2005a) | **86.84** | 30.03 | 44.63 |

**Table 3**
Morpho Challenge 2010 experimental results for German.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 47.92 | **53.84** | **50.71** |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | 57.79 | 32.42 | 41.54 |
| Base Inference (Lignos 2010) | 66.38 | 35.36 | 46.14 |
| Iterative Compounding (Lignos 2010) | 62.13 | 34.70 | 44.53 |
| Aggressive Compounding (Lignos 2010) | 59.41 | 37.21 | 45.76 |
| Morfessor Baseline (Creutz and Lagus 2002) | **82.80** | 19.77 | 31.92 |
| Morfessor CatMAP (Creutz and Lagus 2005a) | 72.70 | 35.43 | 47.64 |

In order to draw a substantive empirical comparison, we performed another set of experiments by running the current approach on only 22K words as the Single Tree Probabilistic Clustering (Can and Manandhar 2012). Results are given in Tables 5, 6, and 7. As shown in the tables, the current model outperforms the previous model even on the smaller data set.

**Table 4**
Morpho Challenge 2010 experimental results for Turkish.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 57.70 | **55.18** | **56.41** |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | 72.36 | 25.81 | 38.04 |
| Base Inference (Lignos 2010) | 72.81 | 16.11 | 26.38 |
| Iterative Compounding (Lignos 2010) | 68.69 | 21.44 | 32.68 |
| Aggressive Compounding (Lignos 2010) | 55.51 | 34.36 | 42.45 |
| Nicolas (Nicolas, Farré, and Molinero 2010) | 79.02 | 19.78 | 31.64 |
| Morfessor Baseline (Creutz and Lagus 2002) | **89.68** | 17.78 | 29.67 |
| Morpho Chain (Narasimhan, Barzilay, and Jaakkola 2015) | 69.25 | 31.51 | 43.32 |
| Morfessor CatMAP (Creutz and Lagus 2005a) | 79.38 | 31.88 | 45.49 |

**Table 5**
Comparison with Single Tree Probabilistic Clustering for English.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | **67.75** | 53.93 | **60.06** |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | 55.60 | **57.58** | 57.33 |

**Table 6**
Comparison with Single Tree Probabilistic Clustering for German.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 33.93 | **65.31** | **44.66** |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | **57.79** | 32.42 | 41.54 |

**Table 7**
Comparison with Single Tree Probabilistic Clustering for Turkish.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 64.39 | **42.99** | **51.56** |
| Single Tree Prob. Clustering (Can and Manandhar 2012) | **72.36** | 25.81 | 38.04 |



### 4.2 Additional Evaluation

For additional experiments, we compare our model with Morpho Chain (Narasimhan, Barzilay, and Jaakkola 2015) based on their evaluation method that differs from the Morpho Challenge evaluation method. Their evaluation method is based on counting the correct segmentation points. For example, if the result segmentation is *booking+s* and the gold segmentation is *book+ing+s*, 1 point is counted. Precision and recall are

**Table 8**
Comparison with Morpho Chain model for English based on Morpho Chain evaluation.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | 67.41 | 62.5 | 64.86 |
| Morpho Chain | **72.63** | **78.72** | **75.55** |

**Table 9**
Comparison with Morpho Chain model for Turkish based on Morpho Chain evaluation.

| System | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Hierarchical Morphological Segmentation | **89.30** | 48.22 | 62.63 |
| Morpho Chain | 70.49 | **63.27** | **66.66** |



calculated based on these matching segmentation points. In addition, this evaluation does not use the hidden gold data sets. Instead, the test sets are created by aggregating the test data from Morpho Challenge 2005 and Morpho Challenge 2010 (as reported in Narasimhan, Barzilay, and Jaakkola [2015]) that provide segmentation points.[3]

We used the same trained models as in our Morpho Challenge evaluation. The English test set contains 2,218 words and the Turkish test set contains 2,534 words.[4]

The English results are given in Table 8 and Turkish results are given in Table 9.

For all systems, Morpho Chain evaluation scores are comparably higher than the Morpho Challenge scores. There are several reasons for this. In the Morpho Challenge evaluation, the morpheme labels are considered rather than the surface forms of the morphemes. For example, *pantolon+u+yla* [with his trousers] and *emel+ler+i+yle* [with his desires] have got both possessive morpheme (*u* and *i*) that is labeled with *POS* and relational morpheme (*yla* and *yle*) labeled with *REL* in common. This increases the total number of points that is computed over all word pairs, and therefore lowers the scores.

Secondly, in the Morpho Chain evaluation, only the gold segmentation that has the maximum match with the result segmentation is chosen for each word (e.g., *yazımıza* has two gold segmentations: *yaz+ı+mız+a* [to our summer] and *yazı+mız+a*; [to our writing]). In contrast, in the Morpho Challenge evaluation all segmentations in the gold segmentation are evaluated. This is another factor that increases the scores in Morpho Chain evaluation. Thus, the Morpho Chain evaluation favors precision over recall. Indeed, in the Morpho Challenge evaluation, the Morpho Chain system has high precision but their model suffers from low recall due to undersegmentation (see Tables 2 and 4).

It should be noted that the output of our system is not only the segmentation points, but also the hierarchical organization of morphological paradigms that we believe is novel in this work. However, because of the difficulty in measuring the quality of hierarchical paradigms, which will require a corresponding hierarchically organized gold data set, we are unable to provide an objective measure of the quality of hierarchical

3 In addition to the hidden test data, Morpho Challenge also provides separate publicly available test data.
4 Because of the unavailability of word2vec word embeddings for German, we were unable to perform Morpho Chain evaluation on this language.

structures learned. We present different portions from the obtained trees in Appendix B (see Figures B.1 and B.2).[5] It can be seen that words sharing the same suffixes are gathered closer to each other, such as *reestablish+ed*, *reclassifi+ed*, *circl+ed*, *uncloth+ed*, and so forth. Secondly, related morphological families gather closer to each other, such as *impress+ively*, *impress+ionist*, *impress+ions*, *impress+ion*.

## 5. Conclusions and Future Work

In this article, we introduce a tree structured Dirichlet process model for hierarchical morphological segmentation. The method is different compared with existing hierarchical and non-hierarchical methods for learning paradigms. Our model learns morphological paradigms that are clustered hierarchically within a forest of trees.

Although our goal in this work is on hierarchical learning, our model shows competitive performance against other unsupervised morphological segmentation systems that are designed primarily for segmentation only. The system outperforms other unsupervised systems in Morpho Challenge 2010 for German and Turkish. It also outperforms the more recent Morpho Chain (Narasimhan, Barzilay, and Jaakkola 2015) system on the Morpho Challenge evaluation for German and Turkish.

The sample paradigms learned show that these can be linked to other types of latent information, such as part-of-speech tags. Combining morphology and syntax as a joint learning problem within the same model can be a fruitful direction for future work.

The hierarchical structure is beneficial because we can have both compact and more general paradigms at the same time. In this article, we use the paradigms only for the segmentation task, and applications of hierarchy learned is left as future work.



## Appendix A. Derivation of the Full Joint Distribution

Let $D = \{s^1, s^2, \ldots, s^N\}$ denote the input data where each $s^j$ is a data item. A particular setting of a table with $N$ customers has a joint probability of:

$$\begin{aligned} p(s^1, \ldots, s^N | \alpha, P) &= p(s^1|\alpha, P)p(s^2|s^1, \alpha, P)p(s^3|s^1, s^2, \alpha, P)\ldots p(s^N|s^1, \ldots, s^{N-1}, \alpha, P) \\ &= \frac{\alpha^L}{\alpha(1+\alpha)(2+\alpha)(N-1+\alpha)} \prod_{i=1}^{L} P(s^i) \prod_{i=1}^{L} (n_{s^i} - 1)! \\ &= \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \alpha^{L-1} \prod_{i=1}^{L} (n_{s^i} - 1)! \prod_{i=1}^{L} P(s^i) \end{aligned} \tag{A.1}$$

For each customer, either a new table is created or the customer sits at an occupied table. For each table, at least one table creation is performed, which forms the second and the fourth factor in the last equation. Once the table is created, factors that represent the number of customers sitting at each table are chosen accordingly, which refers to the third factor in the last equation. All factors with $\alpha$ are aggregated in the first factor in terms of a Gamma function in the last equation.

5 Some of the full trees are given at `http://web.cs.hacettepe.edu.tr/~burcucan/TreeStructuredDP`.

## Appendix B. Sample Tree Structures

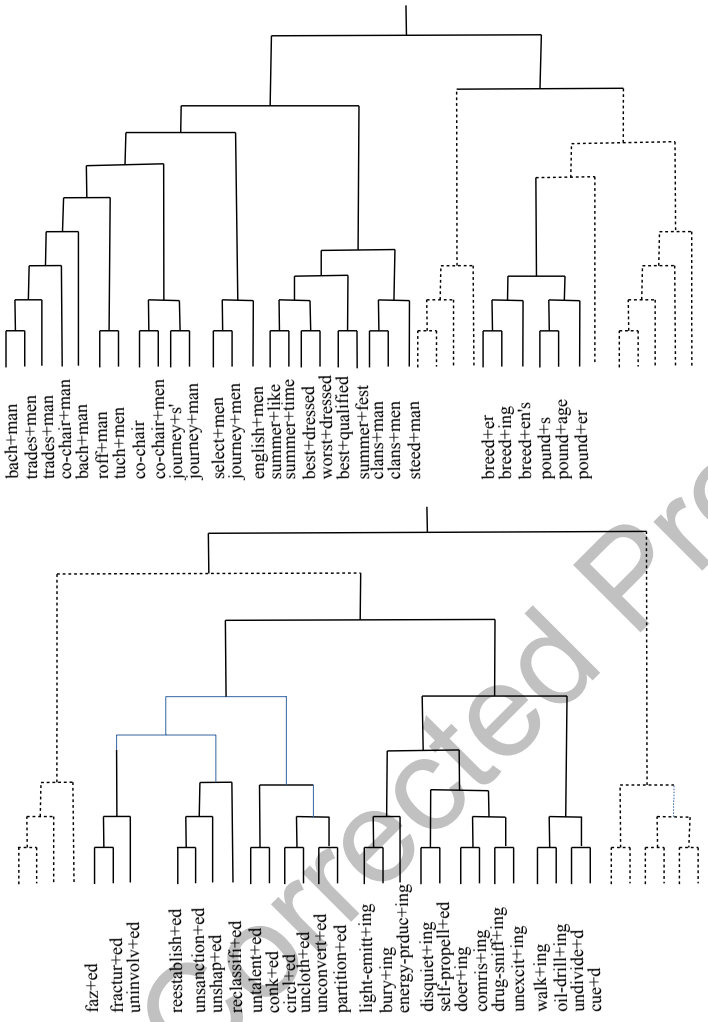





**Figure B.1**
Sample hierarchies in English.

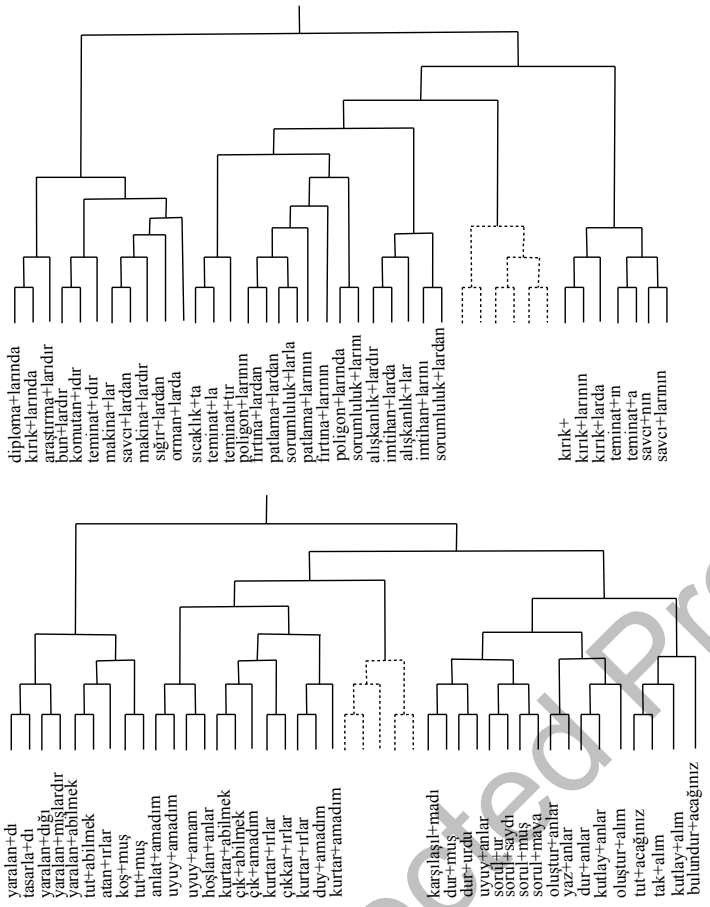



**Figure B.2**
Sample hierarchies in Turkish.



## Acknowledgments

This research was supported by TUBITAK (The Scientific and Technological Research Council of Turkey) grant number 115E464. We thank Karthik Narasimhan for providing their data sets and code. We are grateful to Sami Virpioja for the evaluation of our results on the hidden gold data provided by Morpho Challenge. We thank our reviewers for critical feedback and spotting an error in our previous version of the article. Their comments have immensely helped improve the article.

## References

Can, Burcu. 2011. Statistical Models for *Unsupervised Learning of Morphology and POS tagging*. Ph.D. thesis, Department of Computer Science, The University of York.

Can, Burcu and Suresh Manandhar. 2010. Clustering morphological paradigms using syntactic categories. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, pages 641–648, Corfu.

Can, Burcu and Suresh Manandhar. 2012. Probabilistic hierarchical clustering of morphological paradigms. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 654–663, Avignon.

Chan, Erwin. 2006. Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth*

*Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, SIGPHON '06, pages 69–78, New York, NY.

Creutz, Mathias and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, pages 21–30, Philadelphia, PA.

Creutz, Mathias and Krista Lagus. 2005a. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005)*, pages 106–113, Espoo.

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

Creutz, Mathias and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions Speech Language Processing*, 43:1–3:34.

Dreyer, Markus and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet Process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 616–627, Edinburgh.

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

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Heller, Katherine A. and Zoubin Ghahramani. 2005. Bayesian hierarchical clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 297–304, Bonn.

Hinton, Geoff. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Kurimo, Mikko, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010a. Morpho Challenge Competition 2005–2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, pages 87–95, Uppsala.

Kurimo, Mikko, Sami Virpioja, Ville T. Turunen, Graeme W. Blackwood, and William Byrne. 2009. Overview and results of Morpho Challenge 2009. In *Proceedings of the 10th Cross-Language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments*, CLEF'09, pages 578–597, Corfu.

Kurimo, Mikko, Sami Virpioja, Ville T. Turunen, Bruno Gólenia, Sebastian Spiegler, Oliver Ray, Peter Flach, Oskar Kohonen, Laura Leppänen, Krista Lagus, Constantine Lignos, Lionel Nicolas, Jacques Farré, and Miguel Molinero. 2010b. Proceedings of the Morpho Challenge 2010 Workshop. Technical Report TKK-ICS-R37, Aalto University.

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

Luo, Jiaming, Karthik Narasimhan, and Regina Barzilay. 2017. Unsupervised learning of morphological forests. *Transactions of the Association of Computational Linguistics*, 5:353–364.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Monson, Christian. 2008. *Paramor: From Paradigm Structure to Natural Language Morphology Induction*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

Monson, Christian, Jaime Carbonell, Alon Lavie, and Lori Levin. 2008. Paramor: Finding paradigms across morphology. In *Lecture Notes in Computer Science - Advances in Multilingual and Multimodal Information Retrieval; Cross-Language Evaluation Forum, CLEF 2007*, pages 900–907, Springer, Berlin.

Narasimhan, Karthik, Regina Barzilay, and Tommi S. Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Nicolas, Lionel, Jacques Farré, and Miguel A. Molinero. 2010. Unsupervised learning of concatenative morphology based on frequency-related form occurrence. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 39–43, Espoo.

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 '09, pages 209–217, Boulder, CO.

Snover, Matthew G., Gaja E. Jarosz, and Michael R. Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 11–20, Philadelphia, PA.

Snyder, Benjamin and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, OH.

Teh, Y. W. 2010. In Dirichlet processes, In *Encyclopedia of Machine Learning*, Springer.

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581.