Methods and Algorithms for Unsupervised Learning of Morphology

Burcu Can and Suresh Manandhar

Department of Computer Science,
University of York, Heslington,
York, YO10 5GH, UK
burcu@cs.york.ac.uk,suresh@cs.york.ac.uk

Abstract. This paper is a survey of methods and algorithms for unsupervised learning of morphology. We provide a description of the methods and algorithms used for morphological segmentation from a computational linguistics point of view. We survey morphological segmentation methods covering methods based on MDL (minimum description length), MLE (maximum likelihood estimation), MAP (maximum a posteriori), parametric and non-parametric Bayesian approaches. A review of the evaluation schemes for unsupervised morphological segmentation is also provided along with a summary of evaluation results on the Morpho Challenge evaluations.

Keywords: unsupervised learning, probabilistic models, morphological segmentation, machine learning of morphology

1 Introduction

Morphology is the study of the internal structure of words. The term *'morphology'* was first introduced by the German linguist August Schleicher in 1859 [66]. Morphology refers to the study of how various sub-word units combine together to form new words through a sequence of rule applications. The sub-word units, called *morphemes*, are the smallest meaning bearing units in a word. For example, the word *interestingly* is made up the morphemes *interest*, *ing*, and *ly*.

Morphological segmentation is the process of analysing a word by identifying its constituent morphemes. As a computational problem, morphological segmentation has been treated both as a supervised and unsupervised machine learning problem. In this paper, we provide a survey of existing approaches to unsupervised learning of morphology. Unsupervised learning of morphology is attractive for several reasons: 1. it is able to accommodate changes in the language and 2. it does not require manually annotated data which makes it particularly suitable for resource-poor languages.

Morphological segmentation and morphological analysis are essential preprocessing tasks for many NLP applications. *Speech recognition* is one such application that benefits from morphological segmentation as using whole word dictionary becomes problematic especially for morphologically rich languages

and use of morphemes (or other sub-word unit) sequences rather than word sequences provides better coverage [21, 2, 46, 6, 51, 64]. *Machine translation* is another field that uses morphological segmentation. Machine translation models either use morphological information within the pre-processing step [13, 33, 25] in order to prepare the text for the translation, or morphological segmentation is employed as a post-processing step to generate the inflected morphological forms of words [56, 47]. *Information retrieval* also benefit from morphological segmentation due to the ambiguity and OOV (out-of-vocabulary) words. Within information retrieval, simple morphological approaches like truncation, stemming, stem generation, or lemmatization are often employed [38, 48, 41, 45]. *Question answering* is another application that benefits from morphological segmentation. In a question answering system, morphological analysis is usually required for extracting questions, as well as for the answers that are retrieved. Similar approaches (i.e. stemming, lemmatization, etc.) to the ones used in information retrieval are adopted in order to obtain morphological information in question answering [7, 3].

In this paper, we categorise unsupervised morphology learning methods into the following types:

- *Letter Successor Variety models*: Harris [39], Hafer and Weiss [36], Dejean [26], Bordag [9, 10]
- *Minimum Description Length based models*: Brent et al. [12], Goldsmith's Linguistica [30, 31], Morfessor Baseline MDL [22], Argamon et al. [1], Kazakov & Manandhar [42, 43]
- *Other deterministic approaches*: Bernhard [5], Neuvel and Fulow [60], Keshava and Pitler [44], Monson et al. [57], Lignos et al. [54], Can and Manandhar [14]
- *Maximum likelihood models*: Morfessor Baseline ML [22], Morfessor Categories ML [23], Probabilistic ParaMor [58]
- *Maximum A-Posteriori models*: Morfessor Categories MAP [24]
- *Bayesian parametric models*: Creutz [19], Poon et al. [62]
- *Bayesian non-parametric models*: Goldwater et al. [32], Can and Manandhar [15], Sirts and Alumäe [67], Dreyer and Eisner [28], Snyder and Barzilay [69]

2 Related work

Hammarström [37] is a survey of the work in morphology learning covering a wide range of work between 1955 and 2006. Hammarström provides a synopsis of the field by categorising the studies into four groups: border and frequency methods that detect the segment boundaries either by investigating the substrings that occur frequently with other adjacent substrings or by using the compression of the frequent long substrings; group and abstract methods that analyse morphologically related words in groups (e.g. paradigms); feature-based methods that see words as consisting of various features; and phonology-based methods that analyse words based on their vowels and consonants. Some prominent examples

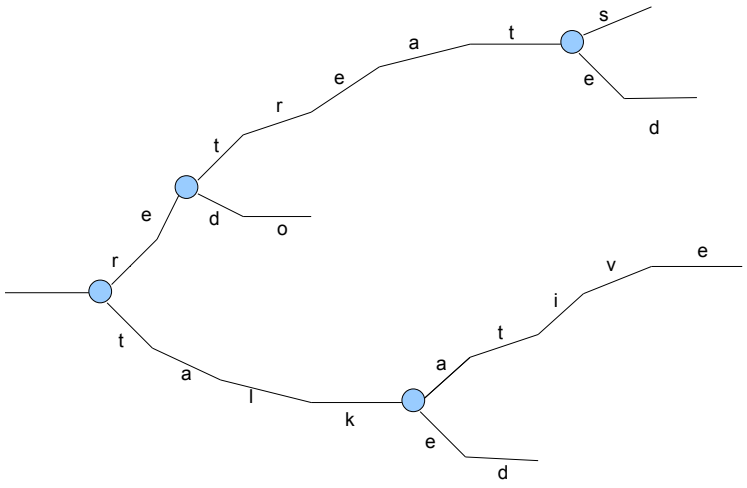

Fig. 1. Word split points in a LSV model

are given for each category. However, the paper does not contain a description of the methods and algorithms employed. It primarily describes the languages that they are tested on, whether the algorithms require any thresholds or parameters to be set by humans, and what the algorithm learns (analysis, paradigms, transducers etc).

Here, our aim is not to survey the same work reviewed by Hammarström from the same perspective. Instead, we aim to focus on the methods and algorithms that have been used for unsupervised morphological segmentation from 1955 till 2013. For this reason, we mainly focus on the methods and algorithms and provide a mathematical overview of the methods from a computational linguistics point of view.

3 Letter Successor Variety (LSV) Models

Harris [39] was the first to introduce the distributional properties of letters within a word and to devise the earliest class of deterministic algorithms for word segmentation. In this model, the potential segmentation points within a word can be characterised by the sharp changes in the number of successors of a letter within a word. For example, a given corpus contains the words *walnut, wall, walks, walked, walking, walk*. The number of letter successors of the prefix *wal-* equals 3, namely, *n, l* and *k*. However, the number of letter successors of *walk* is 4, namely, *s, e, i* and *$* (denoting the word boundary). Harris calls the number of letters that can follow each letter in a word as *successor variety*. Similarly, the letters that precede other letters is called *predecessor variety*.

To determine potential split points, a letter tree (i.e. a trie) is constructed. An example of a letter successor tree is given in Fig. 1. In this example, *re-* is a potential prefix whereas *-s*, *-ed* and *-ing* are potential suffixes on the tree. Harris chooses a cutoff value manually. However, the cutoff value must be chosen carefully. If it is too small, then words are oversegmented. In contrast, if the cutoff is too big, then most true segments are missed.

The successor counts are applied to all words in the corpus to find morpheme boundaries. For example, the procedure may choose *-ing* as a morpheme. Subsequently, all words that precede *-ing* are considered as stems. However, this is problematics since this will cause the model to segment that do not contain *-ing* as a morpheme such as *sing, string, spring, cling, etc.*

Despite this, many researchers have followed the idea of using statistical properties of letter successors and predecessors to identify potential split points. Hafer and Weiss [36] improve the original idea by using the entropy of the successors and predecessors instead of using raw counts. The letter successor entropy (LSE) of a prefix w is defined as follows:

$$LSE(w) = \sum_{c \in \Sigma} -\frac{f(w_c)}{f(w)} \log_2 \frac{f(w_c)}{f(w)} \tag{1}$$

where Σ is the alphabet, $f(w_c)$ is the number of word entries in the corpus that have prefix w followed by the letter c, and $f(w)$ is the total number of the word entries that begin with w and can be followed with any letter.

Mopheme boundaries typically have high LSE and using it improves detection of real morpheme boundaries from non-boundaries that have lower entropies even though both may have the same letter successor counts.

Dejean [26] improves upon Harris's method by dividing the process into 3 different phases. In the first phase, a morpheme dictionary is constructed by using the letter successor variety technique and choosing only the high frequency morphemes. In the second phase, the words in the corpus are segmented using the morpheme dictionary to generate more morphemes. In the final phase, the corpus is analysed by using the morpheme dictionary. For example, given the words *lights, lighting, lighted, lightly, lightness, lightest, lighten.* In the first phase, the most frequent morphemes are selected such that *-s, -ing, -ed, -ly* that have a higher LSV frequency than a given threshold value. In the second phase *-ness, -est,* and *-en* are captured by segmenting the words *lightness, lightest,* and *lighten.* Finally, the entire corpus is morphologically analysed using the combined morpheme dictionary *-s, -ing, -ed, -ly, -ness, -est, -en.*

For example, the words *started, startled, startling* are segmented as *start+ed, start+led, start+ling* in Harris's approach, whereas in Dejean's approach once the morphemes *-ed* and *-ing* are captured, the words are correctly segmented giving *start+ed, startl+ed, startl+ing.*

Bordag [9] does not use any global LSV cutoff value to segment all the words according to the same threshold. Instead, a local LSV value to segment words that are contextually similar is used. The contextual similarity is intended to group words that are syntactically similar. Thus, the idea is to identify syn-

Table 1. Local LSV scores of the word *early* [9].

input word:	e	a	r	l	y
final score:		1.0	0.1	1.0	2.0

Table 2. Local LSV scores of the word *clearly* [9].

input word:	c	l	e	a	r	l	y
final score:		0.4	1.2	0.1	0.4	13.4	4.6

tactically similar words such as subclasses of *adjectives*, *verbs* etc. and choose a different *local* LSV cutoff value for each subclass. With this method, orthographically similar words such as *early* and *clearly* are analysed independently since they tend to be contextually different.

Bordag uses the combination of local LSV weights, the inverse bigram weights, in addition to the original LSV score to obtain a combined score. A cutoff threshold is chosen for the combined score. The scores for *ear-ly* (1.2), *clear-ly* (13.4) permit distinguishing the two cases easily (see Table 1 and Table 2).

Bordag [9, 10] uses the segmentations produced by the local LSV method to train a classifier. Bordag places the morpheme segmentations on a Patricia trie [59] classifier with their frequencies in order to generalise the results for novel words. An example Patricia trie trained by Bordag [10] is given in Figure 2. If a novel word is to be analysed, the *trie* is searched from the root until the correct branch in the trie is found which gives a split for the word. For example, for the novel word *strong*, the trie gives 0.4 by looking at the *root* node only. However, for the novel word *strongly*, the trie gives 0.66 by looking at the *earl* node. Using *tries* helps to handle exceptions as well. For example, a trie with the words *clear+ly*, *strong+ly* and *early* can classify hundreds of words ending with *-ly*, but still remembers one exception which is *early*.

4 Minimum Description Length (MDL) Based Models

According to the MDL principle, the best description of data or the best hypothesis is the one that leads to the best compression of the data. In order to find the best compression of data, the regularities in data need to be captured, as stated by Grünwald [34]:

> "[The MDL Principle] is based on the following insight: any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally." Grünwald [34]

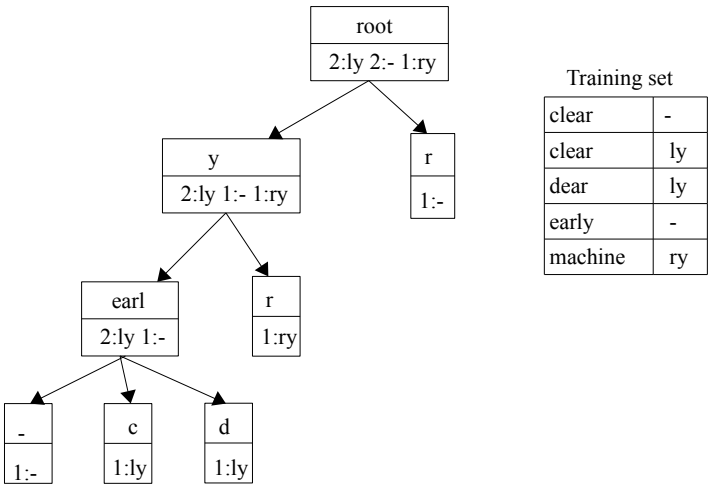

Fig. 2. A sample Patricia trie trained on the training set that contains *clear, clearly, dearly, early,* and *machinery* [10].

From a Bayesian perspective, MDL can be viewed as a prior on the model M:

$$\begin{aligned}\arg\max_{\mathbf{M}} p(M|D) &= \arg\max_{\mathbf{M}} \log_2 p(M|D) \\ &= \arg\min_{\mathbf{M}} [-\log_2 p(M|D)] \\ &= \arg\min_{\mathbf{M}} -log_2 \frac{p(D|M)p(M)}{p(D)} \\ &\propto \arg\min_{\mathbf{M}} -\log_2 [p(D|M)p(M)]\end{aligned}$$

Hence, maximising the posterior probability of a model M given data D is equivalent to minimising the description length of the model times the model likelihood. Equivalently, MDL can be thought as an information theoretic regularisation prior within a MAP estimation model.

Brent et al. [12] encodes the stems and suffixes as binary codes and the encodings are kept in tables (see Tables 3, 4, and 5). The most frequent stems and suffixes are encoded with shorter encodings. The Shannon-Fano (SF) coding [12] is used in order to find the optimal length of each code word. The description length (DL) in bits for the SF coding for a morpheme, m, can be approximated with the negative binary logarithm of its relative frequency:

$$DL(m) = -log_2(freq(m)) \tag{2}$$

Table 3. Input Words

walk	referral
walks	refer
walked	refers
walking	dump
referred	dumps
referring	preferential

Table 4. Stem Table

stem	code
walk	1
referr	2
refer	3
dump	4
preferenti	5

Table 5. Suffix Table

suffix	code
ϵ	1
s	2
ed	3
ing	4
al	5

Table 6. Encoded Words

stem	suffix	stem	suffix
00	00	01	110
00	01	100	00
00	100	100	01
00	101	101	00
01	100	101	01
01	101	1100	110

$$p(M) = \sum_{m \in M} DL(m) \tag{3}$$

A key problem with the approach is that searching through all possible models is not practical. For example, the number of the possible splits of a given text is equal to the product of the lengths of all words in the text. Instead of searching all possible splits of a given text, some heuristics such as first finding the suffix table and then searching for the stem table are employed in Brent's approach.

Linguistica [30, 31] is another system that is based on MDL. In addition to using stem and affix codebooks, Linguistica employs *signature*s to encode the data. A *signature* represents the inner structure of a list of words that have similar inflective morphology. Thus their model consists of: a stem list, an affix list, and a signature list (see Figure 3).

The signature list contains only pointers to stems and affixes[30] and can be thought as an optimal encoding of the signature list. The probability of a segmentation $w = t + f$ is given by:

$$p(w = t + f|\sigma) = p(\sigma)p(t|\sigma)p(f|\sigma) \tag{4}$$

where $p(\sigma)$ is the empirical frequency of the signature σ (normalised); and $p(t|\sigma)$, $p(f|\sigma)$ are the empirical stem, and suffix frequencies (normalised) given the signature σ.

In terms of description length, the size of a word becomes the sum of the size of the pointer to its signature, stem, and affix. For the size, inverse logarithm is used as given in Equation 2. The description length of a corpus is computed through all words in the corpus.

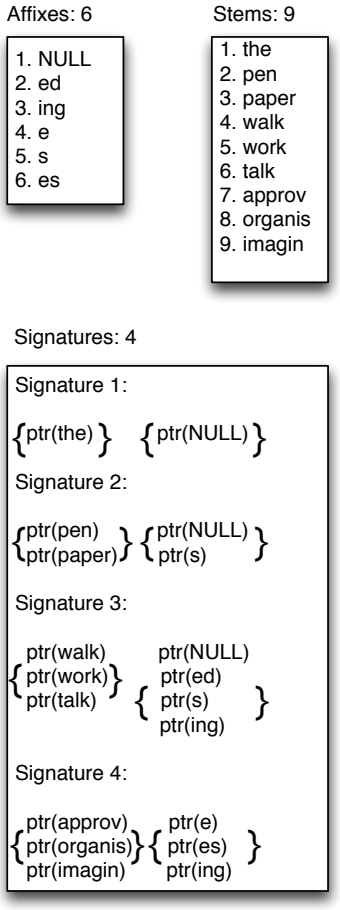

Fig. 3. A sample morphology from Linguistica, that can generate the words: *the, pen, pens, paper, papers, walk, walked walking, walks, work, worked, working, works, talk, talked, talking, talks, approve, approves, approved, organise, organises, organised, imagine, imagines, imagined.*

In order to compute the length of the model, the lengths of all lists are added up:

$$DL(M) = DL(T) + DL(F) + DL(\Sigma) \tag{5}$$

where T is the stem list, F is the suffix list, and Σ is the signature list in the model. Here, the length of each list is the length of each item in the list plus the number of occurrences of each item in the list. Therefore, the description length

of a stem list becomes:

$$DL(T) = \log_2(|T|) + \sum_{t \in T} len(t) \tag{6}$$

where $\log_2(|T|)$ computes the information needed for the number of items in the stem list and $len(t)$ is the number of bits needed for the stem t, which is computed as follows:

$$len(t) = |t| \ log_2 26 \tag{7}$$

where $|t|$ is the number of letters in the stem t by considering a language with 26 letters. The length of a list of affixes is calculated analogously.

In order to calculate the length of a signature list, the length of a pointer has to be determined since the signatures only keep the pointers to the stems, affixes, and other signatures. The length of a pointer to a stem t, suffix f, and signature σ are computed as follows respectively:

$$\log \frac{|W|}{freq(t)}, \log \frac{|W|}{freq(f)}, \log \frac{|W|}{freq(\sigma)}$$

where $|W|$ is the number of words in the corpus and $freq()$ gives the number of occurrences of the given segment in the corpus.

Goldsmith also defines a recursive segmentation procedure that segments words with multiple split points. A flag for each stem is placed in the stem list to determine whether the stem is a simple stem or a complex stem with a triple pointer to a signature, stem, and affix. This modification in the definition of a stem enables the analysis of words such as *[organis-ation]-s* where the stem *organis-ation* is decoded as a complex stem that consists of a pointer to a signature which includes the stem *organis* and the affix *-ation*.

Morfessor Baseline defines the total cost as follows:

$$\begin{aligned} Cost &= DL(D) + DL(M) \\ &= \sum_{i \in D} -\log p(m_i) + \sum_{j \in M} len(m_j) \end{aligned} \tag{8}$$

where m_i denotes the morphemes and $p(m_i)$ denotes the maximum likelihood estimate of the morpheme m_i. The maximum likelihood estimate of a morpheme m_i is the number of token count for m_i divided by the total number of token counts in the corpus. Here the corpus is generated by morphemes in the model. Hence, the length of a corpus is computed by the maximum likelihoods of the morphemes. Morfessor Baseline deploys a recursive segmentation where each discovered morpheme is analysed recursively as long as it improves the cost. The method does not make use of signatures like Linguistica, instead a single *codebook* is used. A similar approach for recursive segmentation has also been used by Argamon et al. [1].

Kazakov & Manandhar [43] develop a hybrid combination of genetic algorithms and inductive logic programming (ILP). A MDL bias is employed within a genetic algorithm by choosing a suitable fitness function that favours codebooks

Table 7. First-order (Prolog) decision-list rules learnt in Kazakov and Manandhar [43]. Exceptions are towards the top and generic rules are towards the bottom.

1. seg(A,B) :-	append([b,l,e,s,s], B, A), !.
2. seg(A,[a,i]) :-	append(_,[a,i],A), !.
3. seg(A,B) :-	append([c,o,m,t,e], B, A),
	append(C,[e,z],A), !.
4. seg(A,B) :-	append([o,r,g,a,n,i,s], B, A),
	append([o,r,g,a,n,i,s, a], C, A), !.
5. seg(A,[a]) :-	append(_, [a], A),
	append(_, [i,r,a], A), !.

with shorter description length. The genetic algorithm generates an initial segmentation. In the following step, segmentation rules are learned from the initial segmentations by employing a first-order decision list learner [55]. The decision-list is able to generalise by learning rules for the segmentation of unseen words. The use of first-order decision lists has two advantages. Firstly, the decision lists easily capture regular expression patterns over which a given segmentation rule applies. Secondly, decision-lists provide a natural mechanism for capturing exceptions since decision-lists are ordered (in terms of priority). Some examples of rules learnt are given in Table 7.

5 Other Deterministic Approaches

We review deterministic methods that do not fall into the categories covered in the previous sections.

Neuvel and Fulow [60] propose an algorithm based on the word-based theory of morphology [29]. In this approach, instead of inducing the morphemes, morphological relations between the words are defined to learn new word forms.

Keshava and Pitler [44] describe an algorithm, RePortS, that is based on using a trie. A forward trie is used for the suffixes, whereas a backward trie is used for the prefixes. Keshava and Pitler define heuristic criteria based on the strings' conditional probabilities on the trie, to identify the suffixes and prefixes by giving them scores. These heuristics are improved by Demberg [27] for handling complex morphology. Lavallée and Langlais [52] use formal analogies to find the relation between 4 word forms, such as {*walking*, *speaker*, *walks*, *speaks*}. However, due to the large search space, such methods can be considered impractical for large lexicons.

Bernhard [5] uses features that combine the length and frequency of morphemes. Stems are generally longer and less frequent than suffixes, whereas suffixes are shorter and more frequent than stems. In order to extract the prefixes and suffixes, the transitional probabilities between substrings are used. First, for each position of the word k the following function is computed:

$$f(k) = \frac{\sum_{i=0}^{k-1} \sum_{j=k+1}^{n} max[p(s_{i,k}|s_{k,j}), p(s_{k,j}|s_{i,k})]}{k(n-k)} \tag{9}$$

Table 8. A sample subgroup of words that contains the stem *hous* and starts with the empty prefix [5].

Words	Suffixes	Potential stems	New suffixes
housekeeping		-ekeeping	
housing	-ing		
household		-ehold	
house's			-e's
house	-e		
housed			

which gives the mean of the maximum transition probabilities for the position k. Here the transition probability $p(s_{i,k}|s_{k,j})$ is estimated as follows:

$$p(s_{i,k}|s_{k,j}) = \frac{f(s_{i,j})}{f(s_{k,j})} \tag{10}$$

where $f(s_{i,j})$ is the frequency of the substring $s_{i,j}$ and the transition probability $p(s_{i,k}|s_{k,j})$ is estimated as follows:

$$p(s_{k,j}|s_{i,k}) = \frac{f(s_{i,j})}{f(s_{i,k})} \tag{11}$$

Local minima of the values of $f(k)$ in a given word correspond to potential morpheme boundaries. Once the morpheme boundaries are found, the longer and less frequent morphemes are chosen as stems and the rest chosen as either prefix or suffix depending on its position. Different words sharing the same stem are compared to find other segments.

ParaMor is a system developed by Monson et al. [57] that discovers candidate suffixes and stems to build paradigms. In their approach, candidate suffixes are any final substrings of words that are found iteratively. Once partial paradigms are built, they are merged by clustering. Eventually, words are segmented by stripping off suffixes that occur in these paradigms. The system is rule based and does not involve a confidence measure. Moreover, the authors combine the results of the ParaMor with Morfessor [20] (named as P+M model). The joint P+M model outperforms other ParaMor variants in several Morpho Challenge evaluations (see Section 11) in terms of f-score.

Lignos et al. [54] employ Base and Transforms model [16] that is based on the discovery of the base and derived forms of words. The discovery is performed through transforms, which are orthographic modifications that are applied on a word to derive another form of the same word. A transform given by (s_1, s_2) removes the suffix s_1 from the word and adds another suffix s_2 to derive another form of the word. Lignos [53] develops an inference procedure that can learn the base form of a word when it is absent in the corpus. The new model handles compounding by decomposing a word into its component words by choosing the highest geometric mean of the component frequencies.

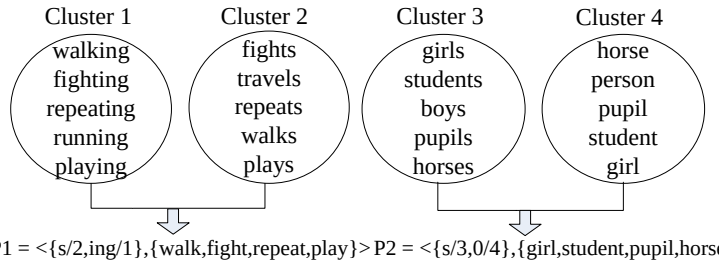

Fig. 4. Paradigm capturing across syntactic categories in the deterministic approach by Can and Manandhar [14].

Can and Manandhar [14] propose a deterministic model that makes use of syntactic categories. Syntactic information and morphology are strongly connected to each other. For example, words ending with *-ly* are generally adverbs, words ending with *-ed* are generally verbs, etc. Syntactic categories are induced using context distribution clustering [17]. Potential suffixes in each syntactic category are ranked by their conditional probability $p(m|c)$ where m denotes the suffix and c denotes the syntactic category. The definition of a morphological paradigm is somewhat different to that of others. Each paradigm consists of a list of morpheme/cluster pairs, m_i/c_i,, and a list of stems, s_i. A paradigm, P, has the form:

$$P =< \{m_1/c_1, \ldots, m_r/c_r\}\{s_1, \ldots, s_k\} >$$

For example, two sample paradigms are:
$P_1 =< \{s/2, ing/1\}\{walk, fight, repeat, play\} >$,
$P_1 =< \{s/3, 0/4\}\{girl, student, pupil, horse\} >$ (see Figure 4).

Suffix pairs that have the maximum number of common stems across two different syntactic categories are merged and a new paradigm is created (see Figure 4). Once the initial morphological paradigms are learnt, they are merged based on their *accuracy* (Acc) as defined below:

$$Acc_1 = \frac{S}{S+N_1},\ Acc_2 = \frac{S}{S+N_2},\ Acc = \frac{Acc_1+Acc_2}{2} \tag{12}$$

where S is the number of common stems between the two paradigms, N_1 is the number of stems that are present in the first paradigm, but absent in the second paradigm (and vice versa for N_2). Higher values of N_1 and N_2 will result in smaller Acc scores and correspondingly lower possibility of merging. Similarly, higher values of S will be preferred for merging. The merging process creates increasingly more general paradigms. The results clearly demonstrate that using syntactic information can help morphological segmentation:

6 Methods based on Maximum Likelihood (ML)

Within Bayesian statistics, Maximum Likelihood (ML) estimation provides, conceptually, the simplest inference procedure for learning models that generalise from data. In morphological segmentation, typically, the model is a probability assignment to possible morphemes. In ML estimation, there is no prior bias towards any model, and the model M that maximises the likelihood function is chosen:

$$M_{ML} = \arg\max_i p(D|M_i) = \arg\max_i log(p(D|M_i)) \tag{13}$$

In Morfessor Baseline ML [22], a model M_i gives a probability distribution over a collection of morphemes. Given such a model, a corpus can be split into its constituent morphemes:

$$log(p(D|M_i)) = \sum_{m \in D} \log p(m|M_i) \tag{14}$$

As this is ML estimation, the model prior is not involved. Initially, words are split with the suffix length drawn from a Poisson distribution. The algorithm employs two *hard* conditions that always reject rare morphemes and single letter morphemes. In that case, word is re-segmented randomly. Otherwise, the segmentation is accepted. An Expectation Maximization (EM) algorithm is employed to find increasingly better segmentations.The inference involves a number of iterations in which 1. the morpheme probabilities are estimated for a given segmentation 2. the text is re-segmented by using the Viterbi algorithm in order to find the segmentation with the lowest cost for each word 3. the segmentation of the word is either accepted or rejected.

The results show that ML approach tends to overspilt when compared to the MDL approach [22]. For example, the word *affectionate* is split as *affecti+on+at+e* in ML approach, where as it is split as *affect+ion+ate* in MDL approach.

Morfessor Categories ML [23] is a Morfessor variant that is also based on ML estimation. In contrast to Morfessor Baseline ML, a hidden Markov model (HMM) is used to assign probabilities to each possible split of a word form. In the model, each morph is emitted from a hidden state that can be interpreted as either prefix, suffix, stem etc. Within a bigram model, the probability of a segmentation of a word w into the morphemes $m_1, m_2, \ldots, m_k$ is computed as follows:

$$p(m_1, m_2, \ldots, m_k|w) = [\prod_{i=1}^{k} p(C_i|C_{i-1})p(m_i|C_i)]p(C_{k+1}|C_k) \tag{15}$$

To learn the HMM transition probabilities, $p(C_i|C_{i-1})$, and the emission probabilities, $p(m_i|C_i)$ (see Figure 5), words are initially segmented by applying the Morfessor Baseline ML [19]. Category membership probabilities $p(C_i|m_i)$ are estimated using a *perplexity* measure. The perplexity measure expresses the predictability of the preceding and following words of a given word. EM is employed

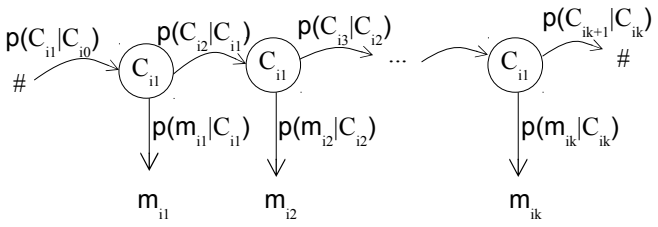

Fig. 5. Transition and emission probabilities of a word w according to Equation 15.

to estimate the probabilities in each iteration after re-tagging the words using the Viterbi algorithm.

Morfessor Categories ML improves upon the Morfessor Baseline for English. Alhough, the Baseline performs slightly better precision, the recall of the Categories ML model is a lot better than the baseline model. In Finnish, for smaller datasets Morfessor Categories ML and Baseline perform on a similar level, however for bigger datasets Morfessor Categories ML performs far better. This work shows that the dependencies between the morphemes play an important role in morphology learning.

Probabilistic ParaMor [58] extends the original ParaMor [57] algorithm by training a finite-stage tagger that will mimic the results of the original ParaMor. The statistical model learns whether each character in a word is the beginning of a new stem or a suffix. The surrounding characters and morpheme-tags (i.e. stem vs suffix) are used as features in the tagger. For the surrounding characters, character unigram, bigram, and trigram morpheme tags are used. For example, in the word *strongly*, the character features for the letter *'o'* consist of *'stro', 'tro', 'ro', 'o', 'on, 'ong', and 'ongly'*. Monson et al. [57] use the averaged perceptron algorithm [18] to train the finite-state tagger. Viterbi search is used for the decoding process. Eventually, the tagger tags each split point within a word as a morpheme boundary or as a continuation of a morpheme. Therefore, the segmentation process is akin to a part-of-speech tagging process.

The probabilistic ParaMor has a higher accuracy compared to the baseline ParaMor. Moreover, the authors combine the results of the baseline ParaMor with Morfessor [20] to train the tagger (named as P+M Mimic model).

7 Methods based on Maximum A-Posteriori (MAP) estimation

In contrast to ML estimation, the *maximum a-posteriori estimation* (MAP) approach allows specifying model prior, $p(M_i)$.

$$M_{MAP} = \arg\max_i p(D|M_i)p(M_i) \tag{16}$$

The MDL models described in Section 6 can be viewed as MAP estimation models with description length (DL) as the the model prior. In this section, we focus on model priors other than those based on DL.

Morfessor Categories MAP [24] employs a first-order HMM in order to model the internal word syntax as given in Figure 5. Morfessor Categories MAP defines a prior for each morpheme using two parameters: meaning and form. The *form* of a morpheme refers to the substructure of the morpheme (made of letters or made of two sub-morphemes). The *meaning* of a morpheme consists of the features such that length, frequency and right/left perplexity of the morpheme. Therefore, the prior probability of a model, M, becomes the combination of the meaning and the form of each morpheme, m_i:

$$p(M) = |M|! \prod_{i=1} M[p(meaning(m_i))p(form(m_i))] \tag{17}$$

The term $|M|!$ accounts for the $|M|!$ possible orderings of the morphs in the model. Thus, the prior favours smaller number of morphemes.

In order to find the model and the corpus segmentation with the minimum cost, a greedy search algorithm is used in Morfessor Categories MAP. In each step, different segmentations for each word are suggested and the one with the maximum probability is chosen. The segmentation of each word is kept in a binary splitting tree. Figure 6 provides an example.

The results for Morfessor Categories MAP are below that of the Morfessor Categories ML. However, the effects of different types features within the prior in MAP models is yet to be explored.

8 Bayesian Parametric Models

Bayesian modelling employs the full form of Bayes' theorem that defines a posterior probability distribution over the parameters in terms of the likelihood $p(D|M)$ and the prior model probability $p(M)$:

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)} \tag{18}$$

Both ML and MAP estimates are point estimates that correspond to the modes of the above distribution. Bayesian modelling introduces a different perspective by representing the estimate in the form of a probability distribution rather than a single point estimate.

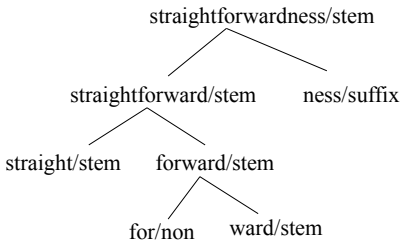

Fig. 6. The hierarchical segmentation of the English word 'straightforwardness' by the Morfessor Categories MAP [24].

One common way to estimate the parameters is to draw random samples from the posterior distribution. Markov Chain Monte Carlo (MCMC) methods are the most common methods employed for sampling from the underlying posterior probability distribution. Samples drawn from the posterior distribution form a Markov chain such that each state is dependent only on the previous state:

$$p(X_{n+1} = x|X_1 = x_1, \ldots, X_n = x_n) = p(X_{n+1} = x|X_n = x_n) \tag{19}$$

The Markov chain converges to a distribution over states, called an equilibrium. Gibbs sampling and Metropolis-Hastings algorithm are the two common MCMC algorithms used for learning segmentation.

Creutz [19] proposes a generative probabilistic model that is intended to overcome the over-segmentation problem in Baseline Morfessor. The proposed model uses prior information on the morpheme lengths and morpheme frequencies, within a generative probabilistic model framework. The model is based on the probabilistic model by Brent [11].

The generative story can be told as follows. The total number of morphemes n is sampled with a uniform distribution. Morpheme lengths l_{m_i} are then drawn from a gamma distribution:

$$p(l_{m_i}) = \frac{1}{\gamma(\alpha)\beta^{\alpha}} l_{m_i}^{\alpha-1} e^{-l_{m_i}/\beta} \tag{20}$$

where α and β are constants, and γ is the gamma function. Once the lengths are drawn, the letters that each morpheme consists of are drawn according to the maximum likelihood of each letter c_j:

$$p(c_j) = \frac{n_{c_j}}{\sum_k n_{c_k}} \tag{21}$$

where n_{c_j} is the frequency of the letter c_j in the corpus, and $\sum_k n_{c_k}$ is the total number of letters in the corpus. Finally, the model/lexicon is created with these morphemes regardless of the order they are created:

$$p(M) = p(n)\ n! \prod_{i=1}^{n} p(m_i) \tag{22}$$

$$p(m_i) = p(l_m) \prod_{j=1}^{l_{m_i}} p(c_j) \tag{23}$$

where n is the number of morphemes in the model, l_m is the length of each morpheme and c_j denotes the letters within morphemes.

Once the model is created, corpus requires to be built by using the morphemes in the model. First, morpheme frequencies are determined by Mandelbrot's correction of Zipf's formula (see Baayen [4]). Finally, the corpus is created according to a particular order by using the inverse of the multinomial:

$$p(Corpus) = \left(\frac{(\sum_{i=1}^{n} f_{m_i})!}{\prod_{i=1}^{n} f_{m_i}!} \right)^{-1} \tag{24}$$

where the numerator is the summation of the morpheme frequencies in the model and the denominator is the product of the factorial of the frequency of each morpheme in the model. The optimal model is searched following a similar recursive search algorithm which is used in the Baseline Morfessor [22]. Results show that the usage of prior information increases the accuracy of the algorithm.

Poon et al. [62] develop a log-linear model where the joint probability between the corpus and all possible segmentations is defined. Since it is not possible to derive all the pairs belonging to the joint probability, a normalisation constant Z is estimated to normalise the joint probability. A few techniques are suggested earlier to compute the normalisation constant. Smith and Eisner [68] apply contrastive estimation by searching around the neighbourhood of the data, whereas Rosenfeld [65] and Poon et al. [63] use sampling to compute the normalisation constant. Poon et al. use both contrastive estimation and sampling to compute the normalisation constant. The neighbourhood is searched by transposing pairs of letters to create invalid words. Gibbs sampling is used to find the optimum segmentation. In the model, also a prior information that is inspired by the MDL model which controls the number of morpheme types in the lexicon and the morpheme tokens in the corpus is used.

9 Bayesian Non-Parametric Models

Bayesian non-parametric models potentially permit an infinite number of parameters to be learnt. In other words, in a non-parametric model, the number of parameters can grow with data. Typically, for example, within morphological segmentation, the number of morpheme classes is not known in advance. Thus,

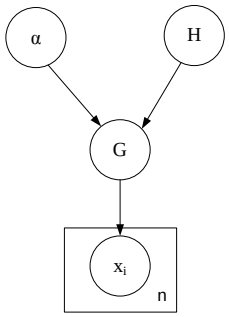

Fig. 7. Plate diagram of a Dirichlet process: $DP(\alpha, H)$ that produces x_i for n times by using the concentration parameter α and the base distribution H.

rather than fixing the number of classes in advance, non-parametric models provide a more realistic and flexible framework to capture the irregularities in data by permitting flexibility in the parameter space.

A well-known approach in Bayesian non-parametric modelling is the *Dirichlet Process*. A Dirichlet process defines a probability distribution over an infinite number of objects [61].

Given data points $x = \{x_1, \ldots, x_N\}$ generated from a Dirichlet process $DP(\alpha, H)$ with a concentration parameter α and a base distribution H (see Figure 7 for the plate diagram):

$$\begin{aligned} x_i &\sim G \\ G &\sim DP(\alpha, H) \end{aligned} \tag{25}$$

the probability of a future observation $x_{N+1} = j$ is given by [8]:

$$\begin{aligned} p(x_{N+1} = j | x, \alpha, H) &= \frac{1}{N+\alpha} \sum_{i=1}^{N} I(x_i = j) + \frac{\alpha}{N+\alpha} H(j) \\ &= \frac{n_j + \alpha H(j)}{N+\alpha} \end{aligned} \tag{26}$$

Here I is an indicator function that outputs 1, if $x_i = j$, otherwise it outputs 0.

This formulation of the Dirichlet process leads to the Chinese Restaurant Process (CRP). Imagine a restaurant that consists of an infinite number of tables with an infinite number of seats at each table where each customer chooses a table and sits down(see Figure 8). At each table, a (possibly) different type of

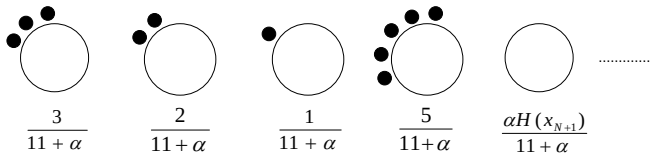

Fig. 8. An illustration of the Chinese Restaurant Process. The new customer x_{N+1} sits at a table which is already occupied with a probability proportional to the number of customers sitting at the table; which is $\frac{3}{11+\alpha}$, $\frac{2}{11+\alpha}$, $\frac{1}{11+\alpha}$, and $\frac{5}{11+\alpha}$ respectively. The customer sits at a table which is empty with a probability proportional with the concentration parameter; which is $\frac{\alpha H(x_{N+1})}{11+\alpha}$.

meal is served. The customer chooses an occupied table with a probability which is proportional to the number of customers who are already sitting at the table, whereas she chooses an empty table with a probability proportional to a defined constant α. Therefore, tables which have a great number of customers attract more customers according to the rich-get-richer principle.

Goldwater et al. [32] introduce a two stage model that extends the Chinese restaurant metaphor, where each table is labelled with a word from a corpus. In their model, initially these labels are generated by a generator component that draws the labels from a multinomial distribution:

$$p(l_k = w) = \sum_{c,t,f} I(w = t + f)p(c_k = c)p(t_k = t|c_k = c)p(f_k = f|c_k = c) \quad (27)$$

where c denotes the class label (which involves a distribution over stems and suffixes), t denotes the stem, and f denotes the suffix that belongs to word w having the label l_k. According to the generative story, first the class label, c_k, is drawn, then the stem, t_k, and suffix, f_k, of the word are drawn conditionally with the class label. Each of these are drawn from multinomial distributions with symmetric Dirichlet priors as follows:

$$\begin{aligned} x_k &\sim Multinomial(\theta) \\ \theta &\sim Dirichlet(\beta) \end{aligned} \quad (28)$$

In the second stage, the actual sequence of words is generated by estimating the frequencies of the words in order to create a power-law distribution. Goldwater et al. [32] use Pitman-Yor process [40][1] for generating the i^{th} word conditioned on all previous words:

$$p(w_i = w|\boldsymbol{w_{-i}}, \boldsymbol{z_{-i}}, \theta) = \sum_{k=1}^{K(z_{-i})} \frac{n_k^{(\boldsymbol{z}_{-i})} - a}{i - 1 + b} I(l_k = w) + \frac{K(\boldsymbol{z}_{-i})a + b}{i - 1 + b}\theta_w \quad (29)$$

[1] The Pitman-Yor process is a generalisation of the Dirichlet process (see [40]).

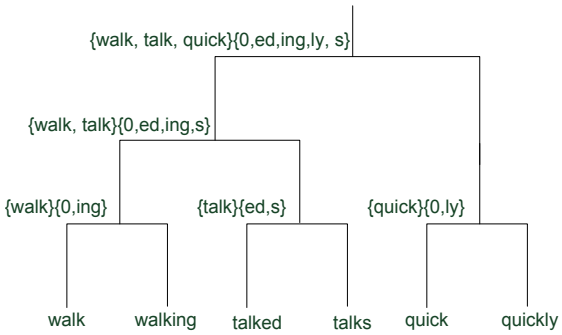

Fig. 9. A sample tree structure.

where z_i denotes the class that generates the ith word, l_k denotes the multinomial distribution over words that belong to the class k, $\boldsymbol{w_{-i}}$ represent the previously generated words, $\boldsymbol{z_{-i}}$ denotes the current seating arrangement, a and b are the parameters of the process, and $K(z_{-i})$ is the number of tables that are occupied. The approach allows different analyses for different tokens of the same word, however only one split point is generated for each word.

Can and Manandhar [15] propose a Dirichlet Process based approach that learns morphological paradigms (see Figure 9). In their approach, morphological paradigms are learned in a hierarchical structure where each node corresponds to a morphological paradigm. The likelihood of data under any subtree is defined recursively by:

$$p(D_k|T_k) = p(D_k)p(D_l|T_l)p(D_r|T_r) \tag{30}$$

where the probability is defined in terms of left T_l and right T_r subtrees. Thus, the likelihood is decomposed recursively until the leaf nodes are reached. The marginal probability is used as prior information since it bears the probability of having the data from the left and right subtrees within a single cluster. The marginal likelihood of words in the node k is defined such that:

$$\begin{aligned} p(D_k) &= p(S_k)p(M_k) \\ &= p(s_1, s_2, \ldots, s_n)p(m_1, m_2, \ldots, m_n) \end{aligned}$$

where $s_1, s_2, \ldots, s_n$ are the stems and $m_1, m_2, \ldots, m_n$ are the suffixes in the node/paradigm k.

Can and Manandhar define two Dirichlet processes to generate stems and suffixes in each node on the hierarchical structure independently:

$$\begin{aligned} G_s|\beta_s, P_s &\sim DP(\beta_s, P_s) \\ G_m|\beta_m, P_m &\sim DP(\beta_m, P_m) \\ s|G_s &\sim G_s \\ m|G_m &\sim G_m \end{aligned}$$

where $DP(\beta_s, P_s)$ denotes a Dirichlet process that generates stems and $DP(\beta_m, P_m)$ denotes a Dirichlet process that generates suffixes, where β_s and β_m are the concentration parameters that determine the number of stem/suffix types in the model. P_s and P_m are the base distributions on the letters that each morpheme consists of, where letters are assumed to be distributed uniformly. Therefore, morphemes having shorter lengths are favoured.

Sirts and Alumäe [67] introduce a non-parametric Bayesian approach for jointly learning morphological segmentation of words along with their part-of-speech tags. Sirts and Alumäe employ a trigram hidden Markov model (HMM) for the part-of-speech tags. The trigram transitions are modelled by hierarchical Dirichlet process (HDP):

$$G^U \sim DP(\alpha^U, H) \tag{31}$$
$$G_j^B \sim DP(\alpha^B, G^U) \tag{32}$$
$$G_{jk}^T \sim DP(\alpha^T, G_j^B) \tag{33}$$

where G_U, G_j^B, and G_{jk}^T are unigram, bigram, and trigram DP's. Unigram DP is used as a base distribution for the bigram DP, where bigram DP is used as a base distribution in the trigram DP. This forms an HDP model. The emission probabilities are modelled with a simple Multinomial-Dirichlet conjugacy. Finally, the segmentations are also modelled as a HDP:

$$G^S \sim DP(\alpha^S, S) \tag{34}$$
$$G_j^{TS} \sim DP(\alpha^{TS}, G^S) \tag{35}$$

where G^S is the common base distribution that is used as a base distribution for the tag-specific DP G_j^{TS} defined for the morphological segments. Here, S is the general base distribution and consists of two components: a geometric distribution over the segment lengths and collapsed Dirichlet-multinomial over character unigrams. Sirts and Alumäe sample tags and morphological segments jointly in their inference algorithm. The results show that learning morphological segments jointly with the part-of-speech tags improve the segmentation. When the tags are fixed and only the morphological segments are learned, it gives lower scores, whereas when both are learned jointly, the results are comparably higher.

Dreyer and Eisner [28] propose an infinite Diriclet mixture model for learning the part-of-speech tag, inflection, lexeme, morphological paradigm of each word in the corpus. For example, *learned* belongs to a verb part-of-speech class, it has past participle inflection, belongs to the lexeme *learn*, and belongs to a morphological paradigm that consists of words *learn, learns, learned, learning*. Dreyer and Eisner construct morphological paradigms via an infinite Dirichlet process mixture model, where each paradigm corresponds to a mixture component having the forms of the same lexeme and word tokens are generated from each paradigm.

10 Evaluation of Morphology Segmentation Algorithms

The evaluation of morphological segmentation requires a gold standard to compare with the suggested analyses, common with most natural language processing tasks. The evaluation process, at first glance, appears straighforward as system generated segmentation need to match the split points in the gold standard. However, especially in morphologically complex languages, additional features such as morphological ambiguity, morphophonology, stem changes etc. can be present. Taking these into consideration, obtaining a gold standard in a range of languages is a demanding task.

Spiegler et al. [70] define the features of a good evaluation metric as:

- Correlating well with other NLP tasks.
- Being computationally easy.
- Being robust.
- Being informative about the strengths and weaknesses of the system.
- Being able to account for the linguistic structure of the language, such as morphophonology, allomorphy, syncretism, and ambiguity.

The evaluation methods for morphological segmentation can be investigated using two categories: intrinsic methods based on a comparison against a gold standard, and, extrinsic methods based on evaluating how the segmentation improves the performance of a NLP task.

Evaluation Using a Gold Standard Segmentation For morphological segmentation, precision, recall and f-score are predominantly used evaluation measures, as in many NLP tasks. F-score is computed as the harmonic mean of precision and recall scores:

$$F\text{-}score = \frac{1}{1/Precision + 1/Recall} \tag{36}$$

Some researchers have used a gold standard consisting of segmentation of all words in a corpus [32, 62]. For this purpose, either a highly accurate morphological analyser is used (for Arabic such as the one by Habash and Rambow [35], or some heuristics are used for the construction of a gold standard (for English, see Goldwater et al. [32].

Instead of using the full corpus for evaluation, Morpho Challenge [50] uses a sampled set of gold standard words for evaluation. In both cases, the gold standard consists of words with their segmentations plus additional morphological information.

For example, given below is example segmentation data from Morpho Challenge. Here morpheme labels represent inflection classes; i.e. plural, past tense form, participle etc.:

ablatives ablative:ablative_A s:+PL
abounded abound:abound_V ed:+PAST

carriages carri:carry_V age:age_s s:+PL
detraction detract:detract_from_V ion:ion_s
entitling entitl:entitle_V ing:+PCP1

To measure precision, from the system generated segmentation of the test words. For each morpheme in the list, another word is found that includes the same morpheme. This will create a word pair list. Finally, word pairs are checked in the gold standard to see whether the pairs share a common morpheme. For each true guess, one point is given. The score is computed by dividing the total number of received points by the number of sampled words. Recall is measured analogously to precision, where the word pairs are sampled from the gold standard, and comparisons are made through the resulting segmentations.

Spiegler and Monson [70] propose a novel evaluation metric called EMMA that does not perform a one-to-one comparison with the gold standard data, but instead finds the maximum match between the suggested segmentations and the gold standard segmentations using an optimal maximum matching (in a bipartite graph).

Evaluation via Other Tasks Another way of evaluating the results of a morphological segmentation is to embed the suggested segmentations into a real world NLP task which utilises the analysed words. In addition to the traditional evaluation metric which is described earlier, Morpho Challenge [49] performs information retrieval and machine translation tasks. In both tasks, words are replaced with the word segmentations. In information retrieval, queries are replaced with their segmentations, whereas in the machine translation task the source language is replaced with its segmentations. Finally, the tasks are evaluated using average precision and *BLEU* score respectively.

11 Evaluation Results in Morpho Challenge

We summarise the Morpho Challenge results for 2007, 2008, and 2009 here to give a better comparison between the models in terms of their accuracy. A wide range of approaches have competed in Morpho Challenge. These have been based on using - Bayesian and frequentist statistics, information theory and heuristics. Depending on the approach taken, the Morpho Challenge evaluations show that some are better in some languages, whereas others are better in other languages.

For English, Bernhard 2 [5] outperforms the other systems in 2007. ParaMor-Morfessor (P+M) [57] outperforms the other systems in 2008. ParaMor-Morfessor (P+M) still outperforms other systems in 2009. For German, ParaMor-Morfessor (P+M) [57] outperforms the other systems in all years. For Turkish Morfessor CatMap. [24] outperforms other systems in 2007. However, Monson ParaMor-Morfessor [58] outperforms others in 2008, and Monson ParaMor-Morfessor [58] Mimic outperforms other systems in 2009.

The results show that hybrid approaches that implement system combinations such as ParaMor-Morfessor (P+M) perform well and there is a still a long way to go to for unsupervised systems.

Table 9. Comparison of methods competed in Morpho Challenge between years 2007 and 2009 for the English language.

Method	2007			2008			2009		
	P	R	F	P	R	F	P	R	F
Bernhard 1 [5]	72.05	52.47	60.72	-	-	-	75.61	57.87	65.56
Bernhard 2 [5]	61.63	60.01	**60.81**	-	-	-	67.42	65.11	66.24
Bordag 5 [9]	59.80	31.50	41.27	-	-	-	-	-	-
Bordag 5a [9]	59.69	32.12	41.77	-	-	-	-	-	-
Can & Manandhar [14]	-	-	-	-	-	-	58.52	44.82	50.76
Lignos [54]	-	-	-	-	-	-	83.49	45.00	58.48
Monson ParaMor [57]	48.46	52.95	50.61	58.50	48.10	52.79	63.32	51.96	57.08
Monson P+M [57]	41.58	65.08	50.74	50.64	63.30	**56.26**	70.09	67.38	**68.71**
Monson P+M Mimic [58]	-	-	-	-	-	-	54.80	60.17	57.36
Morfessor CatMap. [24]	82.17	33.08	47.17	82.17	33.08	47.17	84.75	35.97	50.50
Morfessor Baseline. [22]	-	-	-	71.93	43.27	54.04	74.93	49.81	59.84

Table 10. Comparison of methods competed in Morpho Challenge between years 2007 and 2009 for the German language.

Method	2007			2008			2009		
	P	R	F	P	R	F	P	R	F
Bernhard 1 [5]	63.20	37.69	47.22	-	-	-	66.82	42.48	51.94
Bernhard 2 [5]	49.08	57.35	52.89	-	-	-	54.02	60.77	57.20
Bordag 5 [9]	60.71	40.58	48.64	-	-	-	-	-	-
Bordag 5a [9]	60.45	41.57	49.27	-	-	-	-	-	-
Can & Manandhar [14]	-	-	-	-	-	-	57.67	42.67	49.05
Monson ParaMor [57]	59.05	32.81	42.19	53.42	38.15	44.51	56.98	42.10	48.42
Monson P+M [57]	51.45	55.55	**53.42**	49.53	59.51	**54.06**	64.06	61.52	**62.76**
Monson P+M Mimic [58]	-	-	-	-	-	-	51.07	57.79	54.22
Morfessor CatMap. [24]	67.56	36.92	47.75	67.56	36.92	47.75	84.75	35.97	50.50
Morfessor Baseline. [22]	-	-	-	80.23	19.22	31.01	81.70	22.98	35.87

12 Conclusions

Morphological analysis has a very long history in natural language processing. Modern work in unsupervised morphological segmentation dates back to the work of Harris in the 1950s.

The primary goal of this paper is to survey the methods and algorithms used for unsupervised morphological segmentation with the goal of robust morphological segmentation without using any tagged data. A wide range of methods have been used for unsupervised morphological segmentation. All current methods approach the problem from slightly different perspectives. Some methods employ a form of clustering to segment morphologically similar words cooperatively, while other methods model the internal word syntax for example by using a sequence model such as a HMM. And, some methods benefit from employing

Table 11. Comparison of methods competed in Morpho Challenge between years 2007 and 2009 for the Turkish language.

Method	2007			2008			2009		
	P	R	F	P	R	F	P	R	F
Bernhard 1 [5]	78.22	10.93	19.18	-	-	-	-		-
Bernhard 2 [5]	73.69	14.80	24.65	-	-	-	-	-	-
Bordag 5 [9]	81.44	17.45	28.75	-	-	-	81.19	23.44	36.38
Bordag 5a [9]	81.31	17.58	28.91	-	-	-	81.06	23.51	36.45
Can & Manandhar. [14]	-	-	-	-	-	-	41.39	38.13	39.70
Monson ParaMor [57]	-	-	-	56.67	39.42	46.50	57.35	45.75	50.90
Monson P+M [57]	-	-	-	51.88	52.10	**51.99**	66.78	57.97	62.07
Monson P+M Mimic [58]	-	-	-	-	-	-	48.07	60.39	**53.53**
Morfessor CatMap. [24]	76.36	24.50	**37.10**	-	-	-	79.38	31.88	45.49
Morfessor Baseline. [22]	-	-	-	-	-	-	89.68	17.78	29.67

syntactic/PoS classes. A wide range of mathematical and algorithmic methods have been employed including Bayesian, frequentist, heuristic and information theoretic methods.

As shown in this review, the literature is rather broad and there has been wide range of approaches adopted. Despite the use of current machine learning algorithms, morphological segmentation and more generally unsupervised morphological analysis remains a challenging unsolved problem. Future research could address non-concatenative morphology, stem alternation, morpheme clustering and morphological transformation rule induction.

References

1. Argamon, S., Akiva, N., Amir, A., Kapah, O.: Efficient unsupervised recursive word segmentation using minimum description length. In: Proceedings of the 20th international conference on Computational Linguistics. COLING '04, Stroudsburg, PA, USA, Association for Computational Linguistics (2004) 1058–1064
2. Arısoy, E., Dutağacı, H., Arslan, L.M.: A unified language model for large vocabulary continuous speech recognition of Turkish. Signal Process. **86** (October 2006) pp. 2844–2862
3. Aunimo, L., Heinonen, O., Kuuskoski, R., Makkonen, J., Petit, R., Virtanen, O.: Question answering system for incomplete and noisy data. In Sebastiani, F., ed.: Advances in Information Retrieval. Volume 2633 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, University of Helsinki Department of Computer Science, Finland (2003) 545–545
4. Baayen, R.: Word Frequency Distributions. Kluwer Academic Publishers (2001)
5. Bernhard, D.: Unsupervised morphological segmentation based on segment predictability and word segments alignment. In: PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes. (2006)
6. Berton, A., Fetter, P., Regel-Brietzmann, P.: Compound words in large-vocabulary German speech recognition systems. In: Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on. Volume 2. (oct 1996) 1165 –1168

7. Bilotti, M.W., Katz, B., Lin, J.: What works better for question answering: Stemming or morphological query expansion? In: Proceedings of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004. (2004)
8. Blackwell, D., MacQueen, J.B.: Ferguson distributions via polya urn schemes. The Annals of Statistics **1** (March 1973) 353–355
9. Bordag, S.: Two-step approach to unsupervised morpheme segmentation. In: Proceedings of 2nd Pascal Challenges Workshop. (2006) 25–29
10. Bordag, S. In: Unsupervised and Knowledge-Free Morpheme Segmentation and Analysis. Springer-Verlag, Berlin, Heidelberg (2008) 881–891
11. Brent, M.R.: An efficient, probabilistically sound algorithm for segmentation and word discovery. Machine Learning **34** (February 1999) pp. 71–105
12. Brent, M.R., Murthy, S.K., Lundberg, A.: Discovering morphemic suffixes a case study in mdl induction. In: Fifth International Workshop on AI and Statistics,Ft. (1995) 264–271
13. Brown, P.F., Della Pietra, V.J., Della Pietra, S.A., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2) (June 1993) 263–311
14. Can, B., Manandhar, S.: Clustering morphological paradigms using syntactic categories. In: Proceedings of the 10th Cross-Language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments. CLEF '09, Berlin, Heidelberg, Springer-Verlag (2009) 641–648
15. Can, B., Manandhar, S.: Probabilistic hierarchical clustering of morphological paradigms. In: EACL. (2012) 654–663
16. Chan, E.: Structures and distributions in morphology learning. PhD thesis, University of Pennsylvania (2008)
17. Clark, A.S.: Inducing syntactic categories by context distribution clustering. In: Proceedings of CoNLL-2000 and LLL-2000. (2000) 91–94
18. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 1–8
19. Creutz, M.: Unsupervised segmentation of words using prior distributions of morph length and frequency. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. ACL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 280–287
20. Creutz, M.: Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition. PhD thesis, Computer and Information Science, University of Technology, Espoo, Finland (2006)
21. Creutz, M., Hirsimäki, T., Kurimo, M., Puurula, A., Pylkkönen, J., Siivola, V., Varjokallio, M., Arisoy, E., Saraçlar, M., Stolcke, A.: Morph-based speech recognition and modeling of out-of-vocabulary words across languages. ACM Trans. Speech Lang. Process. **5** (December 2007) pp. 1–29
22. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes. In: Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6. MPL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 21–30
23. Creutz, M., Lagus, K.: Induction of a simple morphology for highly-inflecting languages. In: Proceedings of the 7th Meeting of the ACL Special Interest Group

in Computational Phonology: Current Themes in Computational Phonology and Morphology. SIGMorPhon '04, Stroudsburg, PA, USA, Association for Computational Linguistics (2004) 43–51
24. Creutz, M., Lagus, K.: Inducing the morphological lexicon of a natural language from unannotated text. In: In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning. (2005) 106–113
25. de Gispert, A., Mariño, J.: On the impact of morphology in English to Spanish statistical mt. Speech Communication **50** (November 2008) pp. 1034–1046
26. Déjean, H.: Morphemes as necessary concept for structures discovery from untagged corpora. In: Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning. NeMLaP3/CoNLL '98, Stroudsburg, PA, USA, Association for Computational Linguistics (1998) 295–298
27. Demberg, V.: A language-independent unsupervised model for morphological segmentation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. (2007) 680–685
28. Dreyer, M., Eisner, J.: 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, Edinburgh, Scotland, UK., Association for Computational Linguistics (July 2011) 616–627
29. Ford, A., Singh, R., Martohardjono, G.: Pace Panini. Peter Lang (1967)
30. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics **27(2)** (2001) pp. 153–198
31. Goldsmith, J.: An algorithm for the unsupervised learning of morphology. In: Natural Language Engineering. Volume 12. (2006) 353–371
32. Goldwater, S., Griffiths, T.L., Johnson, M.: Interpolating between types and tokens by estimating power-law generators. In: Advances in Neural Information Processing Systems 18, Cambridge, MA, MIT Press (2006)
33. Goldwater, S., McClosky, D.: Improving statistical mt through morphological analysis. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 676–683
34. Grünwald, P.: A tutorial introduction to the minimum description length principle. In: Advances in Minimum Description Length: Theory and Applications, MIT Press (2005)
35. Habash, N., Rambow, O.: Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 573–580
36. Hafer, M.A., Weiss, S.F.: Word segmentation by letter successor varieties. Information Storage and Retrieval **10**(11-12) (1974) pp. 371 – 385
37. Hammarstrm, H.: A survey and classification of methods for (mostly) unsupervised learning of morphology. In: NODALIDA 2007, the 16th Nordic Conference of Computational Linguistics, Tartu, Estonia, 25-26 May 2007, NEALT (2007)
38. Harman, D.: How effective is suffixing. Journal of the American Society for Information Science **42(1)** (1991) pp. 7–15
39. Harris, Z.S.: From phoneme to morpheme. Language **31**(2) (1955) pp. 190–222
40. Ishwaran, H., James, L.F.: Generalized weighted chinese restaurant processes for species sampling mixture models. Statistica Sinica **13** (2003) 2003

41. Järvelin, K., Pirkola, A.: Morphological processing in mono- and cross-lingual information retrieval. In Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., Yli-Jyrä, A., eds.: Inquiries into Words, Constraints and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday, Stanford, California, CSLI Publications (2005) 214–226
42. Kazakov, D.: Unsupervised learning of naive morphology with genetic algorithms. In: ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks, Prague. (1997) 105–112
43. Kazakov, D., Manandhar, S.: Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. In: Machine Learning. (2001) 43–121
44. Keshava, S., Pitler, E.: A simpler, intuitive approach to morpheme induction. In: In PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes. (2006) 31–35
45. Kettunen, K., Kunttu, T., Järvelin, K.: To stem or lemmatize a highly inflectional language in a probabilistic ir environment? Journal of Documentation **61**(4) (2005) pp. 476–496
46. Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., Stolcke, A.: Morphology-based language modeling for conversational Arabic speech recognition. Computer Speech & Language **20**(4) (October 2006) pp. 589–608
47. Kristina Toutanova, Hisami Suzuki, A.R.: Applying morphology generation models to machine translation. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Columbus, Ohio, USA, Association for Computational Linguistics (2008) 514–522
48. Krovetz, R.: Viewing morphology as an inference process. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '93, New York, NY, USA, ACM (1993) 191–202
49. Kurimo, M., Lagus, K., Virpioja, S., Turunen, V.: Morpho challenge 2010. `http://research.ics.tkk.fi/events/morphochallenge2010/` (June 2011)
50. Kurimo, M., Virpioja, S., Turunen, V.: Proceedings of the morpho challenge 2010 workshop. In: Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology. SIGMORPHON '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 87–95
51. Larson, M., Willett, D., Khler, J., Rigoll, G.: Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In: International Conference on Spoken Language Processing. (2000) 945–948
52. Lavallée, J.F., Langlais, P.: Morphological acquisition by formal analogy. In: Working Notes for the CLEF 2009 Workshop. (September 2009)
53. Lignos, C.: Learning from unseen data. In Kurimo, M., Virpioja, S., Turunen, V., Lagus, K., eds.: Proceedings of the Morpho Challenge 2010 Workshop, Aalto University, Espoo, Finland (2010) 35–38
54. Lignos, C., Chan, E., Marcus, M.P., Yang, C.: A rule-based unsupervised morphology learning framework. In: Working Notes for the CLEF 2009 Workshop. (September 2009)
55. Manandhar, S., Deroski, S., Erjavec, T.: Learning multilingual morphology with clog. In Page, D., ed.: Inductive Logic Programming. Volume 1446 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1998) 135–144
56. Minkov, E., Toutanova, K., Suzuki, H.: Generating complex morphology for machine translation. In: Proceedings of the 45th Annual Meeting of the Association

of Computational Linguistics, Prague, Czech Republic, Association for Computational Linguistics (June 2007) 128–135
57. Monson, C., Carbonell, J., Lavie, A., Levin, L.: Paramor: Finding paradigms across morphology. In: Advances in Multilingual and Multimodal Information Retrieval. Volume 5152 of Lecture Notes in Computer Science. Springer Berlin/Heidelberg (2008) 900–907
58. Monson, C., Hollingshead, K., Roark, B.: Probabilistic ParaMor. In: Proceedings of the 10th CLEF conference on Multilingual information access evaluation: text retrieval experiments. CLEF '09 (September 2009)
59. Morrison, D.R.: Patricia - practical algorithm to retrieve information coded in alphanumeric. Journal of the ACM **15** (October 1968) pp. 514–534
60. Neuvel, S., Fulop, S.A.: Unsupervised learning of morphology without morphemes. In: Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6. MPL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 31–40
61. Orbanz, P., Teh, Y.W.: Bayesian nonparametric models. In: Encyclopedia of Machine Learning. (2010) 81–89
62. Poon, H., Cherry, C., Toutanova, K.: 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, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 209–217
63. Poon, H., Domingos, P.: Joint unsupervised coreference resolution with Markov logic. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '08, Stroudsburg, PA, USA, Association for Computational Linguistics (2008) 650–659
64. Roeland Ordelman, A.V.H., Jong, F.D.: Compound decomposition in Dutch large vocabulary speech recognition. In: Proceedings of Eurospeech 2003. (2003) 225–228
65. Rosenfeld, R.: A whole sentence maximum entropy language model. In: Proceedings of the IEEE Workshop on Speech Recognition and Understanding. (1997)
66. Schleicher, A.: Zur Morphologie der Spreche. Volume 1(7). Mémoires de l'Académie Impériale des Sciences de St. Pétersburg Series VII, St. Pétersburg (1859)
67. Sirts, K., Alumäe, T.: A hierarchical dirichlet process model for joint part-of-speech and morphology induction. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL HLT '12, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 407–416
68. Smith, N.A., Eisner, J.: Contrastive estimation: training log-linear models on unlabeled data. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 354–362
69. Snyder, B., Barzilay, R.: Unsupervised multilingual learning for morphological segmentation. In: Proceedings of ACL-08: HLT, Columbus, Ohio, Association for Computational Linguistics (June 2008) 737–745
70. Spiegler, S., Monson, C.: Emma: A novel evaluation metric for morphological analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING). (August 2010)