INDUCING THE MORPHOLOGICAL LEXICON OF A NATURAL LANGUAGE FROM UNANNOTATED TEXT

Mathias Creutz and Krista Lagus

Neural Networks Research Centre, Helsinki University of Technology,
P.O.Box 5400, FIN-02015 Espoo, FINLAND,
{mathias.creutz, krista.lagus}@hut.fi

ABSTRACT

This work presents an algorithm for the unsupervised learning, or induction, of a simple morphology of a natural language. A probabilistic maximum a posteriori model is utilized, which builds hierarchical representations for a set of morphs, which are morpheme-like units discovered from unannotated text corpora. The induced morph lexicon stores parameters related to both the "meaning" and "form" of the morphs it contains. These parameters affect the role of the morphs in words. The model is implemented in a task of unsupervised morpheme segmentation of Finnish and English words. Very good results are obtained for Finnish and almost as good results are obtained in the English task.

1. INTRODUCTION

With the emergence of large amounts of textual data in several languages the prospects for designing algorithms that are capable of acquiring language in an unsupervised manner from data seem more and more promising. Also due to the large amounts of data available there is an increasing need for minimally supervised natural language processing systems.

In writing systems where word boundaries are not explicitly marked, word segmentation is the first necessary step for any natural language processing task dealing with written text. Languages employing such writing systems comprise, e.g., Chinese and Japanese.

Existing algorithms for automatic word segmentation usually rely on man-made lexicons (e.g., Sproat et al., 1996) or they are trained on pre-segmented text (e.g., Teahan et al., 2000). However, there are also a number of data-driven algorithms that work more or less without supervision and induce, from nothing more than raw text, a plausible segmentation of a text into words, e.g., de Marcken, 1996; Kit and Wilks, 1999; Brent, 1999; Yu, 2000; Ando and Lee, 2000; Peng and Schuurmans, 2001.

Even if word boundaries are marked in the writing system of a language, words may consist of lengthy sequences of morphemes. Morphemes have been defined in linguistic theory as the smallest meaning-bearing units as well as the smallest elements of syntax (Matthews, 1991). Therefore, morphemes can conceivably be very useful in artificial language production or understanding as well as in applications, such as speech recognition (Siivola et al., 2003; Hacioglu et al., 2003), machine translation and information retrieval.

Automatic segmentation of words into morphemes or morpheme-like units can take place using unsupervised, data-driven morphology inducing algorithms (e.g., Déjean, 1998; Goldsmith, 2001; Creutz and Lagus, 2002; Creutz, 2003; Creutz and Lagus, 2004), which resemble algorithms for word segmentation.

Some of the word and morpheme segmentation algorithms have drawn inspiration from the works of Z. Harris, where a word or morpheme boundary is suggested at locations where the predictability of the next letter in a letter sequence is low (Déjean, 1998; Ando and Lee, 2000). However, in this work we will investigate methods that are aimed not only at the most accurate segmentation possible, but *additionally learn a representation of the language* in the data. Typically, the representation, which is induced from the data, consists of a lexicon of words or morpheme-like units. A word or morpheme segmentation of the text is then obtained by choosing the most likely sequence of words or morphemes contained in the lexicon.

We present a new model and algorithm for simple morphology induction based on previous work (Creutz and Lagus, 2002; Creutz, 2003; Creutz and Lagus, 2004). The latest method as well as previous versions will hereafter be referred to as the *Morfessor* family. The motivations behind the new model will be discussed in Section 2 and the mathematical formulation follows in Section 3. Section 4 reports on experiments carried out on the unsupervised morpheme segmentation of Finnish and English words, while Section 5 concludes the paper.

2. REPRESENTATION OF LEXICAL INFORMATION

The models addressed in this work are formulated either using the Minimum Description Length (MDL) (Rissanen, 1989) or maximum a posteriori (MAP) framework. These two approaches are essentially equivalent, which has been demonstrated, e.g., by Chen, 1996. The aim is to find the optimal balance between *accuracy* of representation and model *complexity*, which generally improves generalization capacity by inhibiting overlearning.

A central question regarding morpheme segmentation is the *compositionality* of meaning and form. If the meaning of a word is transparent in the sense that it is the "sum

of the meaning of the parts", then the word can be split into the parts, which are the morphemes, e.g., English 'foot+print', 'joy+ful+ness', 'play+er+s'. However, it is not uncommon that the form does consist of several morphemes, which are the smallest elements of syntax, but the meaning is not entirely compositional, e.g., English 'foot+man' (male servant wearing a uniform), 'joy+stick' (control device), 'sky+scrap+er' (very tall building).

2.1. Composition and perturbation

De Marcken (1996) proposes a model for unsupervised language acquisition, which involves two central concepts: *composition* and *perturbation*. Composition means that an entry in the lexicon is composed of other entries, e.g., 'joystick' is composed of 'joy' and 'stick'. Perturbation means that changes are introduced that give the whole a unique identity, e.g., the meaning of 'joystick' is not exactly the result of the composition of the parts. This framework is similar to the class hierarchy of many programming languages, where classes can modify default behaviors that are inherited from superclasses.

Among other things, de Marcken applies his model in a task of unsupervised word segmentation of a text, where the blanks have been removed. As a result, hierarchical segmentations are obtained, e.g., for the phrase 'for the purpose of': [[f[or]][[t[he]][[[p[ur]][[[po]s]e]][of]]]]. The problem here from a practical point of view is that there is no way of determining which level of segmentation corresponds best to a conventional word segmentation. On the coarsest level the phrase works as an independent "word" ('forthepurposeof'). On the most detailed level the phrase is shattered into individual letters.

2.2. Baseline morph segmentation

In the so called Recursive MDL method by Creutz and Lagus (2002) and the follow-up (Creutz, 2003) words in a corpus are split into segments called *morphs*. We hereafter call these methods the *Morfessor Baseline* algorithm. The Morfessor Baseline model is also described in a technical report (Creutz and Lagus, 2005) and software implementing it is publicly available[1]. The Baseline is rather similar to some unsupervised word segmentation algorithms, e.g., Brent, 1999; Kit and Wilks, 1999; Yu, 2000. In the Morfessor Baseline, a lexicon of morphs is constructed, so that it is possible to form any word in the corpus by the concatenation of some morphs. Each word in the corpus is then rewritten as a sequence of morph pointers, which point to entries in the lexicon. The aim is to find the optimal lexicon and segmentation, i.e., a set of morphs that is concise, and moreover gives a concise representation for the corpus.

A consequence of this kind of approach is that frequent word forms remain unsplit, whereas rare word forms are excessively split. This follows from the fact that the most concise representation is obtained when any frequent word is stored as a whole in the lexicon (e.g., English 'having', 'soldiers', 'states', 'seemed'), whereas rarely occurring words are better coded in parts (e.g., 'or+p+han', 's+ed+it+ious', 'vol+can+o'). There is no proper notion of compositionality in the model, because frequent strings are usually kept together whereas rare strings are split. In contrast with the model proposed by de Marcken, the lexicon is flat instead of hierarchical, which means that any possible inner structure of the morphs is lost.

[1] `http://www.cis.hut.fi/projects/morpho/`

arvo	n	lisä	vero	ttoma	sta
value	of	addition	tax	-less	from

Figure 1. Morpheme segmentation of the Finnish word 'arvonlisäverottomasta' ("from [something] exclusive of value added tax").

2.3. Learning inflectional paradigms

Goldsmith (2001) assumes a restrictive word structure and his algorithm *Linguistica* splits words into one stem followed by one (possibly empty) suffix. Also prefixes are allowed. Sets of stems and suffixes are grouped together into so called signatures, which are inflectional paradigms discovered from the training corpus. While Linguistica in principle handles stem+suffix-like compositional structure better than the Morfessor Baseline method, it also has the advantage of modeling a simple morphotactics (word-internal syntax). For instance, Linguistica is much less likely to suggest typical suffixes in the beginning of words, a mistake occasionally made by the Baseline (e.g., 'ed+ward', 's+urge+on', 's+well'). Unfortunately, Goldsmith's model poorly suits highly-inflecting or compounding languages, where words can consist of possibly lengthy sequences of morphemes with an alternation of stems and suffixes. Figure 1 shows an example of such a Finnish word.

2.4. Morphotactics for highly-inflecting languages

The so called Categories model (hereafter called the *Morfessor Categories-ML* model) presented by Creutz and Lagus (2004) remedies many of the shortcomings of the Morfessor Baseline and Goldsmith's Linguistica. The model is a maximum likelihood (ML) model that functions by reanalyzing a segmentation produced by the Morfessor Baseline algorithm. The Categories-ML algorithm operates on data consisting of word types, i.e., one single occurrence is picked for every distinct word form occurring in the corpus. Words are represented as Hidden Markov Models (HMM:s), where there are three latent morph categories: prefixes, stems, and suffixes (and an additional temporary "noise" category). The categories emit morphs (word segments) with particular probabilities. There is context-sensitivity corresponding to a simple morphotactics due to the transition probabilities between the morph categories. Stems can alternate with prefixes and suffixes, but there are some impossible category sequences: Suffixes are not allowed in the beginning and prefixes at the end of words. Furthermore, it is impossible to move directly from a prefix to a suffix without passing through a stem.

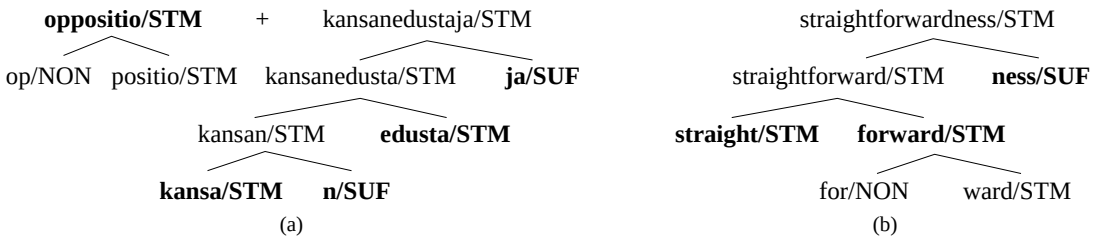

Figure 2. The hierarchical segmentations of (a) the Finnish word 'oppositiokansanedustaja' ("MP of the opposition") and (b) the English word 'straightforwardness' (obtained by the Categories-MAP model; see Section 2.5 for details). Additionally, every morph is tagged with a category, namely the most likely category for that morph in that context.

Compositionality is handled in an approximative manner: If a morph in the lexicon consists of other morphs that are present in the lexicon (e.g., 'seemed = seem+ed'), a split is forced (with some restrictions), and the redundant morph is removed from the lexicon. If on the other hand, a word has been shattered into many short fragments, these are under some conditions considered to be "noise". Noise morphs are removed by joining them with their neighboring morphs, which hopefully creates a proper morph (e.g., 'or+p+han' becomes 'orphan').

Even though the Morfessor Categories-ML algorithm performs rather well, the formulation of the model is somewhat ad hoc. Moreover, the data fed to the algorithm consist of a corpus vocabulary, i.e., a word type collection where all duplicate word forms have been removed. This means that all information about word frequency in the corpus is lost. If we wish to draw parallels to language processing in humans, this is an undesirable property, because word frequency seems to play an important role in human language processing. Baayen and Schreuder (2000) refer to numerous psycholinguistic studies that report that high-frequency words are responded to more quickly and accurately than low-frequency words in various experimental tasks. This effect is obtained regardless whether the words have compositional structure or not.

2.5. Functionality and elegance

The new model proposed in this work, called *Categories-MAP*, draws inspiration from de Marcken (1996). A hierarchical lexicon is induced, where a morph can either consist of a string of letters or of two submorphs, which can recursively consist of submorphs. As in the Categories-ML model, words are represented by HMM:s and there are the same four morph categories: prefix (PRE), stem (STM), suffix (SUF), and non-morpheme (NON). Whether a morph is likely to function as any of these categories is determined by its "meaning", which corresponds to features collected about the *usage* of the morph within words. The model is expressed in a maximum a posteriori (MAP) framework, where the likelihood of category membership follows from the usage parameters through prior probability distributions.

Figure 2 shows hierarchical representations obtained for the Finnish word 'oppositiokansanedustaja' ("member of parliament of the opposition") and the English word 'straightforwardness'. The Categories-MAP model utilizes information about word frequency: The English word has been frequent enough in the corpus to be included in the lexicon as an entry of its own. The Finnish word has been less frequent and is split into 'oppositio' ("opposition") and 'kansanedustaja' ("member of parliament"), which are two separate entries in the lexicon induced from the Finnish corpus. Frequent words and word segments can thus be accessed directly, which is economical and fast. At the same time, the inner structure of the words is retained in the lexicon, because the morphs are represented as the concatenation of other (sub)morphs, which are also present in the lexicon: The Finnish word can be bracketed as [op positio][[[kansa n] edusta] ja] and the English word as [[straight [for ward]] ness].

Not all morphs in the lexicon need to be "morpheme-like" in the sense that they represent a meaning. Some morphs correspond more closely to syllables and other short fragments of words. The existence of these non-morphemes makes it possible to represent some longer morphs more economically, e.g., the Finnish 'oppositio' consists of 'op' and 'positio' ("position"), where 'op' has been tagged as a non-morpheme and 'positio' as a stem. Sometimes this helps against the *oversegmentation* of rather rare words. When for instance, a new name must be memorized, it can be constructed from shorter familiar fragments without breaking it down into individual letters. For example, in one of the English experiments the name 'Zubovski' occurred twice in the corpus and was added to the morph lexicon as 'zubov/STM + ski/NON'.

In the task of morpheme segmentation, the described data structure is very useful. While de Marcken had no means of knowing which level of segmentation is the desired one, we can expand the hierarchical representation to the *finest resolution that does not contain non-morphemes*. In Figure 2 this level has been indicated using a bold-face font. The Finnish word is expanded to 'oppositio + kansa + n + edusta + ja' (literally "opposition + people + of + represent + -ative"). The English word is expanded into 'straight + forward + ness'. The morph 'forward' is not expanded into 'for + ward', because 'for' is tagged as a non-morpheme in the current context.

3. MATHEMATICAL FORMULATION OF THE MODEL & SEARCH ALGORITHM

We aim at finding the optimal lexicon and segmentation, i.e., a set of morphs that is concise and gives a concise

representation for the corpus. The maximum a posteriori (MAP) estimate to be maximized is thus:

$$\arg\max_{\textit{lexicon}} P(\textit{lexicon} \mid \textit{corpus}) = \arg\max_{\textit{lexicon}} P(\textit{corpus} \mid \textit{lexicon}) \cdot P(\textit{lexicon}). \quad (1)$$

The search for the configuration that yields the highest overall probability involves several steps, which are explained briefly in Section 3.6. The calculation of $P(\textit{lexicon})$ and $P(\textit{corpus} \mid \textit{lexicon})$ is described below.

3.1. Probability of the morph lexicon

The lexicon consists of M distinct morphs (i.e., morph types). The probability of coming up with a particular set of M morphs making up the lexicon can be written as:

$$P(\textit{lexicon}) = M! \cdot \prod_{i=1}^{M} \left[P(\textit{meaning}(\mu_i)) \cdot P(\textit{form}(\mu_i)) \right]. \quad (2)$$

Here the probability of each morph μ_i has been divided into two separate parts: one for the "meaning" of μ_i and one for the "form" of μ_i. These terms are discussed in Sections 3.3 (form) and 3.4 (meaning) below. The factor $M!$ is explained by the fact that there are $M!$ possible orderings of a set of M items and the lexicon is the same regardless of the order in which the M morphs emerged.

3.2. Probability of the segmented corpus

A first-order Hidden Markov Model is utilized in order to model a simple morphotactics or word-internal syntax. The probability of the corpus, when a particular lexicon and morph segmentation is given, takes the form:

$$P(\textit{corpus} \mid \textit{lexicon}) = \prod_{j=1}^{W} \left[P(C_{j1} \mid C_{j0}) \prod_{k=1}^{n_j} \left[P(\mu_{jk} \mid C_{jk}) \cdot P(C_{j(k+1)} \mid C_{jk}) \right] \right]. \quad (3)$$

The product is taken over the W words in the corpus (token count), which are each split into n_j morphs. The k^{th} morph in the j^{th} word, μ_{jk}, has been assigned a category C_{jk}, and the probability of the morph is the probability that the morph is emitted by the category, written as $P(\mu_{jk} \mid C_{jk})$. Additionally there are transition probabilities $P(C_{j(k+1)} \mid C_{jk})$ between the categories, where C_{jk} denotes the category assigned to the k^{th} morph in the word, and $C_{j(k+1)}$ denotes the category assigned to the following, or $(k+1)^{\text{th}}$, morph. The transition probabilities comprise transitions from a special word boundary category to the first morph in the word, $P(C_{j1} \mid C_{j0})$, as well as the transition from the last morph to a word boundary, $P(C_{j(n_j+1)} \mid C_{jn_j})$.

3.3. Form of a morph

The probability of the form of the morph μ_i depends on whether the morph is represented as a string of letters (4a) or as the concatenation of two submorphs (4b):

$$P(\textit{form}(\mu_i)) = \begin{cases} (1 - P(\sigma)) \prod_{j=1}^{\textit{length}(\mu_i)} P(c_{ij}). & (4a) \\ P(\sigma) P(C_{i1} \mid \sigma) P(\mu_{i1} \mid C_{i1}) P(C_{i2} \mid C_{i1}) P(\mu_{i2} \mid C_{i2}). & (4b) \end{cases}$$

$P(\sigma)$ is the probability that a morph has substructure, i.e., the morph consists of two submorphs. $P(\sigma)$ is estimated from the lexicon by dividing the number of morphs having substructure by the total number of morphs.

In (4a), $P(c_{ij})$ is the probability of the j^{th} letter in the i^{th} morph in the lexicon. The last letter of the morph is the *end-of-morph character*, which terminates the morph. The probability distribution to use for the letters in the alphabet can be estimated from the corpus (or the lexicon).

Equation 4b resembles Equation 3, where the probability of the corpus is given. $P(C_{i1} \mid \sigma)$ is the probability that the first morph in the substructure is assigned the category C_{i1}. $P(C_{i2} \mid C_{i1})$ is the transition probability between the categories of the first and second submorphs. $P(\mu_{i1} \mid C_{i1})$ and $P(\mu_{i2} \mid C_{i2})$ are the probabilities of the submorphs μ_{i1} and μ_{i2} conditioned on the categories C_{i1} and C_{i2}. The transition and morph emittance probabilities are the same as in the probability of the corpus (Eq. 3).

3.4. Features related to the meaning of a morph

It is a common view that the meaning of words (or morphs) is reflected directly in how they are used. In this work, some parameters related to the usage of morphs in words are collected. These parameters are both properties of the morph itself and properties of the context it typically appears in. The typical usage of the morph is stored in the lexicon together with the form, i.e., the symbolic realization, of the morph (see Equation 2).

The set of features used in this work for defining the meaning of a morphis very limited. As properties of the morph itself, we count the *frequency* of the morph in the segmented corpus and the *length* in letters of the morph. As "distilled" properties of the context the morph occurs in, we consider its intra-word *right* and *left perplexity*. As a consequence, the probability of the meaning of the morph μ_i, $P(\textit{meaning}(\mu_i))$, is the product of the prior probabilities of the frequency, length, right and left perplexity of μ_i.

Note, however, that the set of *possible* features is very large: The typical set of morphs that occur in the context of the target morph could be stored. Typical syntactic relations of the morph with other morphs could be included. The size of the context could vary from small to big, revealing different aspects of the meaning of the morph, from fine-grained syntactic categories to broader semantic or topical distinctions.

3.4.1. Frequency

Frequent and infrequent morphs generally have different semantics. Frequent morphs can be function words and affixes as well as common concepts. The meaning of

frequent morphs is often ambiguous as opposed to rare morphs, which are predominantly names of persons, locations and other phenomena.

The morph emission probabilities $P(\mu_{jk} \mid C_{jk})$ (Eq. 8) depend on the frequency of the morph in the training data. The probability of the lexicon is affected by the following prior for the frequency distribution of the morphs:

$$P(\textit{freqs}) = 1/\binom{N-1}{M-1} = \frac{(M-1)!(N-M)!}{(N-1)!}, \quad (5)$$

where N is the total number of morph *tokens* in the corpus, which equals the sum of the frequencies of the M morph *types* that make up the lexicon. Equation 5 is derived from combinatorics: As there are $\binom{N-1}{M-1}$ ways of choosing M positive integers that sum up to N, the probability of one particular frequency distribution of M frequencies summing to N is $1/\binom{N-1}{M-1}$. Note that the probability of every frequency of every morph in the lexicon is given by one equation instead of computing separate probabilities for every morph frequency.

3.4.2. Length

The length of a morph affects the probability of whether the morph is likely to be a stem or belong to another morph category. Stems often carry semantic (as opposed to syntactic) information. As the set of stems is very large in a language, stems are not likely to be very short morphs, because they need to be distinguishable from each other.

Creutz (2003) defines a prior distribution for morph length. However, in this work, no such explicit prior is used, because the length of a morph can be deduced from the representation of the form of the morph in the lexicon (Section 3.3).

3.4.3. Left and right perplexity

The left and right perplexity give a very condensed image of the immediate context a morph typically occurs in. Perplexity serves as a measure for the predictability of the preceding or following morph.

Grammatical affixes mainly carry syntactic information. They are likely to be common "general-purpose" morphs that can be used in connection with a large number of other morphs. We assume that a morph is likely to be a prefix if it is difficult to predict what the following morph is going to be. That is, there are many possible right contexts of the morph and the right perplexity is high. Correspondingly, a morph is likely to be a suffix if it is difficult to predict what the preceding morph can be and the left perplexity is high. The right perplexity of a target morph μ_i is calculated as:

$$\textit{right-ppl}(\mu_i) = \Big[\prod_{\nu_j \in \text{ right-of}(\mu_i)} P(\nu_j \mid \mu_i) \Big]^{-\frac{1}{f_{\mu_i}}}. \quad (6)$$

There are f_{μ_i} occurrences of the target morph μ_i in the corpus. The morph tokens ν_j occur to the right of, immediately following, the occurrences of μ_i. The probability distribution $P(\nu_j \mid \mu_i)$ is calculated over all such ν_j. Left perplexity can be computed analogously.

As a reasonable probability distribution over the possible values of right and left perplexity, we use *Rissanen's universal prior* for positive numbers (Rissanen, 1989):[2]

$$P(n) \approx 2^{-\log_2 c - \log_2 n - \log_2 \log_2 n - \log_2 \log_2 \log_2 n - \ldots}, \quad (7)$$

where the sum includes all positive iterates, and c is a constant, about 2.865.

3.5. Morph emission probabilities

This section describes how the properties related to the meaning of a morph are translated into the emission probabilities $P(\mu_{jk} \mid C_{jk})$, which are needed in Eq. 3 and 4b. First, Bayes' formula is applied:

$$\begin{aligned} P(\mu_{jk} \mid C_{jk}) &= \frac{P(C_{jk} \mid \mu_{jk}) \cdot P(\mu_{jk})}{P(C_{jk})} \quad (8) \\ &= \frac{P(C_{jk} \mid \mu_{jk}) \cdot P(\mu_{jk})}{\sum_{\forall \mu_{j'k'}} P(C_{jk} \mid \mu_{j'k'}) \cdot P(\mu_{j'k'})}. \end{aligned}$$

The category-independent probabilities $P(\mu_{jk})$ are maximum likelihood estimates, i.e., they are computed as the frequency of the morph μ_{jk} in the corpus divided by the total number of morph tokens.

The tendency of a morph to be assigned a particular category, $P(C_{jk} \mid \mu_{jk})$, (e.g., the probability that the English morph 'ness' functions as a suffix) is derived from the parameters related to the use of the morph in words. A graded threshold of *prefix-likeness* is obtained by applying a sigmoid function to the right perplexity of a morph:

$$\textit{prefix-like}(\mu_{jk}) = \big(1 + \exp[-a \cdot (\textit{right-ppl}(\mu_{jk}) - b)]\big)^{-1}. \quad (9)$$

The parameter b is the perplexity threshold, which indicates the point where a morph μ_{jk} is as likely to be a prefix as a non-prefix. The parameter a governs the steepness of the sigmoid. The equation for suffix-likeness is identical except that left perplexity is applied instead of right perplexity.

As for stems, we assume that the *stem-likeness* of a morph correlates positively with the *length* in letters of the morph. A sigmoid function is employed as above, which yields:

$$\textit{stem-like}(\mu_{jk}) = \big(1 + \exp[-c \cdot (\textit{length}(\mu_{jk}) - d)]\big)^{-1}. \quad (10)$$

where d is the length threshold and c governs the steepness of the curve.

Prefix-, suffix- and stem-likeness assume values between zero and one, but they are no probabilities, since they usually do not sum up to one. A proper probability distribution is obtained by first introducing the *non-morpheme* category, which corresponds to cases where *none* of the proper morph classes is likely. Non-morphemes are typically short, like the affixes, but their right

[2] Actually Rissanen defines his universal prior over all *non-negative* numbers and he would write $P(n-1)$ on the left side of the equation. Since the lowest possible perplexity is one, we do not include zero as a possible value in our formula.

and left perplexities are low, which indicates that they do not occur in a sufficient number of different contexts in order to qualify as a pre- or suffix. The probability that a segment is a non-morpheme (NON) is:

$$P(\text{NON} \mid \mu_{jk}) = [1 - \textit{prefix-like}(\mu_{jk})] \cdot [1 - \textit{suffix-like}(\mu_{jk})] \cdot [1 - \textit{stem-like}(\mu_{jk})]. \quad (11)$$

Then the remaining probability mass is distributed between prefix, stem and suffix (proportionally to the square of the prefix-, stem- and suffix-likeness values).

Finally, if the morph consists of submorphs, its category membership probabilities are affected by the category tagging of the submorphs. This prevents conflicts between the syntactic role of a morph itself and its substructure. Only if either submorph has been tagged as a non-morpheme, no dependencies apply, because non-morphemes are considered as mere sound patterns without a syntactic (or semantic) function. Otherwise the following dependencies are used: Stems need to consist of at least one (sub)stem (PRE + STM, STM + STM, or STM + SUF). Suffixes can only consist of other suffixes. A morph consisting of two suffixes has a fair chance of being tagged as a suffix itself, even though its left perplexity is not very high. Prefixes are treated analogously to the suffixes.

3.6. Search algorithm

The search for the most probable Categories-MAP segmentation takes place using the following greedy search algorithm. In an attempt to avoid local maxima of the overall probability function, steps of resplitting and rejoining morphs are alternated. The steps are briefly described in the sections to follow.

1. Initialization of a segmentation.
2. Splitting of morphs.
3. Joining of morphs using a bottom-up strategy.
4. Splitting of morphs.
5. Resegmentation of corpus using Viterbi algorithm and re-estimation of probabilities until convergence.
6. Repetition of Steps 3–5 once.
7. Expansion of the morph substructures to the finest resolution that does not contain non-morphemes.

3.6.1. Initialization

First, the Morfessor Baseline algorithm is used for producing an initial segmentation of the words into morphs. No morph categories are used at this point. Upon termination of the search, the segments obtained are tagged with category labels according to the equations in Section 3.5. From this point on, the full Categories-MAP model is used as it has been formulated mathematically above.

Producing a reasonably good initial segmentation was observed to be important, apparently due to the greedy nature of the Morfessor Categories-MAP search algorithm. When a bad initial segmentation was used in preliminary experiments the result was clearly poorer.

3.6.2. Splitting of morphs

The morphs are sorted into order of increasing length. Then every possible substructure of a morph is tested, i.e., every possible split of a morph into two submorphs. The most probable split (or no split) is chosen. Additionally different category taggings of the morphs are tested. Since there are transition probabilities, changes affect the context in which a morph occurs. Therefore, the same morph is evaluated separately in different contexts, and as a result different representations can be chosen in different contexts.

There are four morph categories plus an additional word boundary category. This implies that there are $(4 + 1) \cdot (4 + 1) = 25$ different combinations of preceding and following category tags. We have chosen to cluster these 25 cases into four different contexts in order to increase the expected number of observations of a particular morph in a particular context. The clustering increases the probability mass of the tested modifications, which increases the probability that the search does not get stuck in suboptimal local maxima. The four contexts are (a) word initial, (b) word final, (c), word initial and final, (d) word internal. A preceding word boundary or prefix makes a context "word initial" in this scheme, whereas a succeeding word boundary or suffix makes a context "word final".

Not all morphs are processed in the same round of morph splitting. At times the splitting of morphs is interrupted. The whole corpus is retagged using the Viterbi algorithm and the probabilities are re-estimated, after which the splitting continues.

3.6.3. Joining of morphs bottom-up

Morphs are joined together to form longer morphs, starting with the most frequent morph bigrams and proceeding in order of decreasing frequency. The most probable alternative of the following is chosen: (i) Keep the two morphs μ_1 and μ_2 separate; (ii) Concatenate the morphs to a new morph μ_0 having no substructure; (iii) Add a higher level morph μ_0 which has substructure and consists of $\mu_1 + \mu_2$. Additionally, different category taggings of the morphs are tested. The same morph bigram is evaluated separately in different contexts, just as in the splitting of morphs above. At times the joining of morphs is interrupted. The whole corpus is retagged using the Viterbi algorithm and probabilities are re-estimated, after which the morph joining continues.

4. EXPERIMENTS

The Categories-MAP algorithm has been evaluated in a morpheme segmentation task, both on Finnish and English data. "Gold standard" segmentations for the words were obtained from *Hutmegs* (Creutz and Lindén, 2004), which contains linguistic morpheme segmentations for 1.4 million Finnish and 120 000 English word forms[3].

The Finnish data consist of prose and news texts from the Finnish IT Centre of Science (CSC) and the Finnish National News Agency. The English data are composed

[3] `http://www.cis.hut.fi/projects/morpho/`

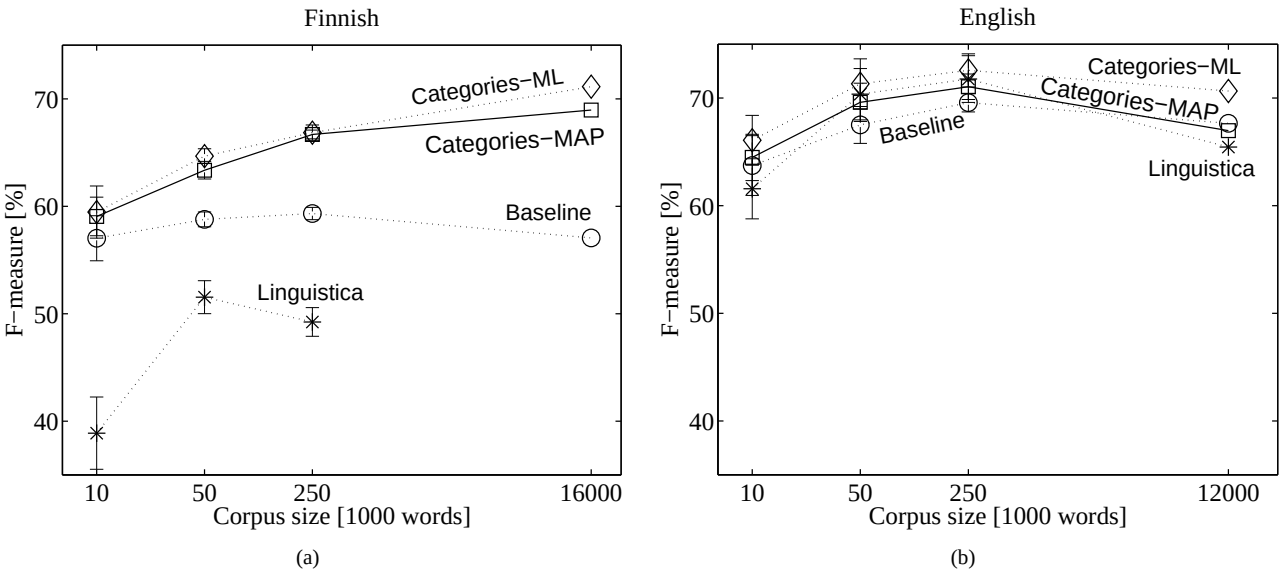

Figure 3. Morpheme segmentation performance of Categories-MAP and three other algorithms on (a) Finnish and (b) English test data. Each data point is an average of 5 runs on separate test sets, with the exception of the 16 million words for Finnish and the 12 million words for English (1 test set). In these cases the lack of test data constrained the number of runs. The standard deviations of the averages are shown as intervals around the data points. There is no data point for Linguistica on the largest Finnish test set, because the program is unsuited for very large amounts of data due to its considerable memory consumption.

of prose, news and scientific texts from the Gutenberg project, the Brown corpus, and a sample of the Gigaword corpus. Evaluations were carried out on data sets containing 10 000, 50 000, 250 000 and 16 million words for Finnish. The same data set sizes were used for English, except for the largest data set, which contained 12 million words. Parameter values (Equations 9 and 10) were set using held-out development sets, which were not part of the final test sets.

As an evaluation metric the *F-measure* is used, which is the harmonic mean of *precision* and *recall* and combines the two values into one:

$$\textit{F-Measure} = 1/[\frac{1}{2}(\frac{1}{\textit{Precision}} + \frac{1}{\textit{Recall}})]. \qquad (12)$$

Precision is the proportion of correct boundaries among all morph boundaries suggested by the algorithm. Recall is the proportion of correct boundaries discovered by the algorithm in relation to all morpheme boundaries in the gold standard. The evaluation is performed on a corpus vocabulary (word types), i.e., each word form (frequent or rare) has equal weight in the evaluation.

4.1. Results

The F-measure of the segmentations obtained on the Finnish and English test sets are shown in Figure 3. The performance of the new Morfessor Categories-MAP algorithm is compared to the performance of the Morfessor Baseline and Categories-ML algorithms as well as Goldsmith's Linguistica[4] (see Section 2). A more detailed comparison of the three older algorithms has been presented in (Creutz and Lagus, 2004).

Figure 3a shows that Categories-MAP performs very well in the morpheme segmentation of Finnish words and it rivals Categories-ML as the best-performing algorithm. For the data sizes 10 000 and 250 000 words the difference between the two is not even statistically significant (T-test level 0.05). For English (Figure 3b), the difference between all the algorithms is overall smaller than for Finnish. Also here Categories-MAP places itself between the best-performing Categories-ML and the Baseline algorithm, except for the largest data set, where Categories-MAP falls slightly below the Baseline. On the English data the difference is statistically significant only between Categories-ML and the lowest-scoring algorithm (Linguistica at 10 000 words; Baseline at 50 000 & 250 000 words).

For English, the achieved F-measure is on the same level as for Finnish, but the advantage of Categories-MAP compared to the simpler Baseline method is less evident. A decrease in F-measure is observed for all four algorithms on the largest English data set. This set contains many foreign words, which may explain the degradation in performance, but a more careful examination of this finding is needed.

4.2. Computational requirements

The Categories-MAP algorithm was implemented as a number of Perl scripts and makefiles. The largest Finnish data set took 34 hours and the largest English set $2\frac{1}{2}$ hours to run on an AMD Opteron 248, 2200 MHz processor. The memory consumption never exceeded 1 GB. The other algorithms were considerably faster, but Linguistica was

[4] `http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/` (December 2003 version)

very memory-consuming.

One can also compare the number of distinct morph types present in the segmentation of the data, a figure reflecting the size of the morph lexicon induced. Out of the algorithms compared, Morfessor Baseline tends to produce a lexicon with the smallest number of entries, while Linguistica produces the largest lexicons. The sizes of the morph inventories discovered by the Morfessor Category models do not differ much from each other: around 110 000 morphs were discovered from the largest Finnish data set, and 50 000 morphs from the largest English set.

5. CONCLUSIONS

In this work, we have demonstrated how the meaning and form of morpheme-like units can be modeled in a morphology induction task and how this model can be used for the morpheme segmentation of word forms. An important feature of the new Morfessor Categories-MAP model is that frequent complex entities have a representation of their own, but the inner structure of these entities is represented as well and can be examined at the desired level of detail.

In the future one might attempt to model non-concatenative phenomena such as sound changes occurring in word stems. So far the modeling of meaning has only been touched upon and could be extended, e.g., one might use semantically richer contextual information, obtained from either longer textual contexts or multimodal data. Moreover, the current model family assumes the existence of distinct, albeit probabilistic categories. In order to develop the model family towards continuous latent representations one might draw inspiration from the conceptual spaces framework proposed by Gärdenfors (2000).

6. ACKNOWLEDGMENTS

We are grateful to the Graduate School of Language Technology in Finland for providing funding for this work. We are also very thankful to the persons sharing stimulating ideas with us, and especially to Krister Lindén and Vesa Siivola as well as the anonymous reviewers for their helpful comments on the manuscript.

References

Ando, R. K. and Lee, L. (2000). Mostly-unsupervised statistical segmentation of Japanese: Applications to Kanji. In *Proc. 6th Applied Natural Language Processing Conference and 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL)*, pages 241–248.

Baayen, R. H. and Schreuder, R. (2000). Towards a psycholinguistic computational model for morphological parsing. *Philosophical Transactions of the Royal Society (Series A: Mathematical, Physical and Engineering Sciences 358)*, pages 1–13.

Brent, M. R. (1999). An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

Chen, S. F. (1996). *Building Probabilistic Models for Natural Language*. PhD thesis, Harvard University.

Creutz, M. (2003). Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proc. ACL'03*, pages 280–287, Sapporo, Japan.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proc. Workshop on Morphological and Phonological Learning of ACL'02*, pages 21–30, Philadelphia, Pennsylvania, USA.

Creutz, M. and Lagus, K. (2004). Induction of a simple morphology for highly-inflecting languages. In *Proc. 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 43–51, Barcelona.

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

Creutz, M. and Lindén, K. (2004). Morpheme segmentation gold standards for Finnish and English. Technical Report A77, Publications in Computer and Information Science, Helsinki University of Technology.

de Marcken, C. G. (1996). *Unsupervised Language Acquisition*. PhD thesis, MIT.

Déjean, H. (1998). Morphemes as necessary concept for structures discovery from untagged corpora. In *Workshop on Paradigms and Grounding in Natural Language Learning*, pages 295–299, Adelaide.

Gärdenfors, P. (2000). *Conceptual Spaces*. MIT Press.

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

Hacioglu, K., Pellom, B., Ciloglu, T., Ozturk, O., Kurimo, M., and Creutz, M. (2003). On lexicon creation for Turkish LVCSR. In *Proc. Eurospeech'03*, pages 1165–1168, Geneva, Switzerland.

Kit, C. and Wilks, Y. (1999). Unsupervised learning of word boundary with description length gain. In *Proc. CoNLL99 ACL Workshop*, Bergen.

Matthews, P. H. (1991). *Morphology*. Cambridge Textbooks in Linguistics, 2nd edition.

Peng, F. and Schuurmans, D. (2001). Self-supervised Chinese word segmentation. In *Proc. Fourth International Conference on Intelligent Data Analysis (IDA)*, pages 238–247. Springer.

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

Siivola, V., Hirsimäki, T., Creutz, M., and Kurimo, M. (2003). Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proc. Eurospeech'03*, pages 2293–2296, Geneva, Switzerland.

Sproat, R., Shih, C., Gale, W., and Chang, N. (1996). A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Teahan, W. J., Wen, Y., McNab, R., and Witten, I. H. (2000). A compression based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3):375–393.

Yu, H. (2000). Unsupervised word induction using MDL criterion. In *Proc. ISCSL*, Beijing.