MORPHEME SEGMENTATION BY OPTIMIZING TWO-PART MDL CODES

Krista Lagus, Mathias Creutz, Sami Virpioja and Oskar Kohonen

Adaptive Informatics Research Centre,
Helsinki University of Technology,
P.O.Box 5400, FIN-02015 TKK, FINLAND
krista.lagus@tkk.fi

1. INTRODUCTION

In many real-world NLP applications, a compact yet representative vocabulary is a necessary ingredient. Words are often thought of as basic units of representation. In highly-inflecting and compounding languages, words can consist of long sequences of meaningful segments, such as prefixes, stems and suffixes: kahvi + n + juo + ja + lle + kin (also for the coffee drinker). Overlooking regularities caused by the common elements accentuates the problem of data sparsity, which is a serious problem for the accurate estimation of statistical language models.

In statistical language modeling the task is to estimate probabilities of word sequences. The state-of-the art approach in applications such as speech recognition, is to model word sequences as Markov chains, i.e. using the n-gram model. However, while it obtains reasonable performance in English, with languages like Finnish the n-gram model runs into serious problems having to do with data sparsity. The reason may be understood by looking at how the vocabulary size increases with corpus size in different languages, as shown in Fig. 1. If complete word forms are the basic linguistic segments, the size of the vocabularies that are needed for NLP applications become very large, especially for highly inflecting and compounding languages. Finding a better segmentation of the linguistic data is therefore useful.

From a linguistic point of view, Finnish frequently employs inflecting (e.g. 'sorme+t', 'finger+s') and compounding (e.g. 'vasen+kätinen', 'left+handed'). It is thus very productive on the word level: any number of new words can easily be produced in this manner by a competent language user. Since many word forms in a sentence are thus rare, obtaining reasonable probability estimates for longer word sequences becomes very hard.

In contrast, if these long, compound and inflected word forms can be split automatically into reasonable segments, then even if the complete compound has not been seen before, each segment may be familiar, and therefore obtain at least a unigram probability estimate that is more accurate than the probability estimate reserved for predicting out-of-vocabulary items.

The financial support from Academy of Finland is gratefully acknowledged.

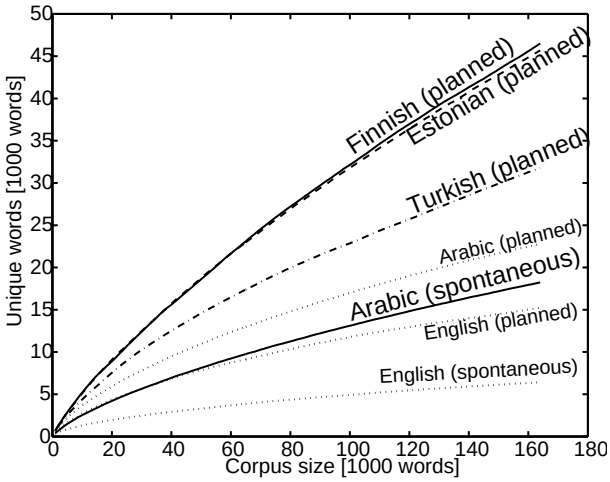

Figure 1. The number of unique words when more text is observed in different languages. 'Planned' refers to written news text, whereas 'spontaneous' consists of transcripts of phone conversations.

There exist linguistic methods and automatic tools for retrieving morphological analyses for words, e.g., based on the two-level morphology formalism [1]. However, these systems require extensive tailoring by linguistic experts for each new language. Moreover, when new words emerge, their morphological analyses must be manually added to the system.

Inspired by the coding philosophy of the Minimum Description Length principle (MDL) by Rissanen [2], we decided to apply MDL to the problem of discovering a segmentation of words into their smaller representative parts. Our hope was that instead of finding, say, syllables, this would lead us to find *meaningful* parts, that is, linguistic morphemes. Moreover, there were interesting similarities between codes found by MDL and properties of natural languages. Natural language can, of course, be viewed as a code for communicating ideas. Many natural languages seem to exhibit the property that frequent words tend to be shorter, while rare words can be arbitrarily long.

Another source of inspiration was an early unsupervised morpheme segmentation method called Linguistica [3], which, while reasonably good for English, made as-

sumptions that rendered it less suitable for morphologically rich languages.

The method we have developed, later called Morfessor, has been described in a series of papers starting with [4]. While we started by deriving the model in the MDL framework, the latest version of the model was expressed in the Maximum A Posteriori (MAP) estimation framework, e.g. [5]. However, the coding philosophy of MDL underlies also the development of the later versions and thus even they can perhaps be understood best from the coding viewpoint.

2. A METHOD FOR MORPHEME SEGMENTATION

We have utilised the two-part MDL coding, or Crude MDL as it was recently called by Grünwald [6][1]. The intuitive coding idea behind the two-part MDL is as follows: Modeling can be viewed as a problem of how to encode a data set efficiently in order to transmit it to a listener with a minimal number of bits. To transmit a data set, one first transmits the model, then the data set by referring to the model. Thus we would like to find the model $\mathcal{M}^*$:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} L(\mathcal{M}) + L(\text{corpus}|\mathcal{M}) \qquad (1)$$

In the case of segmenting words into morphemes, the model can simply consist of the collection of unique morphemes, and a pointer assigned for each. The corpus is then transmitted by sending a sequence of pointers, each representing a morpheme as it occurs in the text. The relationship between coding length L (the costs) and probabilities is $L(\mathcal{M}) = -\log P(\mathcal{M})$.

The model can be thought to consist of a lexicon of word segments called morphs, and a grammar that contains morphotactic information about how the morphs may be combined, in other words, word-internal syntactic dependencies. However, in the basic approach we disregard morphotactic dependencies and make the simplifying assumption that the model is simply the lexicon, which contains an entry for each distinct morph. The corpus (data) can then be encoded as a sequence of pointers to the morph lexicon.

In the Baseline version of the Morfessor, the probability of coming up with a particular set of M morphs $\mu_1 \ldots \mu_M$ making up the lexicon can be written as:

$$\begin{aligned} P(\mathcal{M}) &= P(\textit{lexicon}) && (2)\\ &= M!P(f_{\mu_1}, \ldots, f_{\mu_M})P(s_{\mu_1}, \ldots, s_{\mu_M}) && (3)\\ &= M!\frac{1}{\binom{N-1}{M-1}} \times \prod_{i=1}^{M}\prod_{j=1}^{l_{\mu_i}} P(c_{ij}). && (4) \end{aligned}$$

where for each morph μ_i we only encode in the lexicon its frequency in the data set (f_{μ_i}) and the string of the morph (s_{μ_i}). The factor $M!$ is explained by the fact that there are $M!$ possible orderings of M morphs, and the lexicon is the same regardless of the ordering. The probability distribution of the morph frequencies corresponds to a *non-informative prior*, where the sum of frequencies N is divided into M positive integers f_{μ_i} (the morph frequencies) so that any distribution of M frequency values summing to N has equal probability. The strings of the morphs are encoded as sequences of l_{μ_i} independent characters c_{ij}. The lengths of the morphs can be encoded implicitly by using one character as a end-of-morph marker.

Finally, to calculate $L(\text{corpus}|\mathcal{M})$, the corpus is thought to be encoded as a sequence of morph pointers. The cost of each pointer is $-\log P(\mu_i)$, where $P(\mu_i)$ is the probability of a morph μ_i occurring in the corpus.

A simple greedy algorithm is applied for finding the morpheme lexicon. Initially, the lexicon contains all the words in the corpus. In one training epoch, the words are picked in a random order, and for each word, a segmentation to two parts is attempted. The two segments are added to the morph lexicon (if not already there), and the word itself is removed from it. If the overall cost diminishes, the proposed change is accepted, otherwise rejected. Furthermore, after the best split has been found, a recursive re-splitting is attempted for each of the obtained parts. Learning is halted when the improvements to the cost obtained in an epoch decrease below a set threshold. A pseudocode of the algorithm can be found, e.g., in [7].

3. RESULTS AND DISCUSSION

Table 1 shows sample segmentations on Finnish and English data (adapted from Table 2 of [7]). While quite often the splits are located at linguistic morpheme boundaries, there are also errors due to disregarding the context of each segment. Consequently, frequent word endings can be erroneously encoded also in the beginnigs of words, as in 's+wing', 'ed+ward'. Furthermore, very rare words that do not share segments with other words, such as some foreign names, may be split even after each letter (since each

Table 1. Morph segmentations learned by Morfessor Baseline for a few Finnish and English words. The Finnish examples are inflections of the word 'hellä' (tender, affectionate). The segmentations happen to be correct, except for the plural marker 'i' that has been attached to the stem in the two last words.

Finnish example	English example
hellä	tender
hellä + ä	tender + er
hellä + än	tender + est
hellä + ksi	tender + hearted + ness
hellä + nä	tender + ize
hellä + sti	tender + ly
helli + ksi	tender + ness
helli + nä	tender + ness + es

[1] While the Refined MDL appears to be theoretically better than the two-part or Crude MDL, we have considered the two-part formulation more useful—it is intuitive, and facilitates choosing a model that captures our understanding of the problem. We warmly encourage the interested reader to show us how the same can be said about the refined MDL formulation.

letter is included in the morph lexicon).

An important benefit compared to manually coded linguistic models is in terms of practical applicability: Morfessor is able to provide rather good segmentations also for unseen words, whereas the linguistic models rely on pre-composed dictionary, and cannot handle new words.

When applied as basic units in n-gram language models for Finnish speech recognition the discovered morphs improve accuracy markedly, when compared to using words or linguistic morphemes as modeling units [8]. The greatest improvement over word-based models seems to come from the fact that there are no longer out-of-vocabulary (OOV) elements in the new data (cf. Fig. 3 in [9]).

In later variants of the model (Morfessor-Cat-ML and Morfessor-Cat-MAP), the aim was to find a way of modeling also the morphotactic dependencies. Therefore, another layer of representation was added, namely a HMM model of the segments with the hidden categories prefix, stem and suffix. The categories are allowed to occur only in certain combinations as restricted by the regular expression $(\text{prefix} * \text{stem} + \text{suffix}*)+$. This has lead to clearly improved segmentation results, when compared to a linguistic gold standard segmentation of words [5]. The later models have been described using the MAP estimation framework, but some of the priors are more understandable from the MDL perspective.

The later models' segmentation performance leaves still some room for improvement from a linguistic point of view. In particular, the models cannot discover allomorphic variation (e.g. 'lla' and 'llä' are different, context-dependent surface forms of a morpheme with roughly the meaning 'on'). In an extension of Morfessor Baseline called Allomorfessor [10] this problem is considered. It is assumed that a linguistic (latent) morpheme can have orthographical variants. The variants are coded using operations (insertions, deletions and replacements) applied on the morpheme, each operation having a coding length.

Recently, we have begun the study on how a similar MDL-based approach could be applied to finding a compact description of longer sequences of text, namely sentence-level utterances [11].

To conclude, the Morfessor method family has been applied with good results to a variety of languages, including Finnish, English, German, Turkish and Arabic. It is being extensively researched and applied in speech recognition systems in Finnish and many other morphologically rich languages (see, e.g., [8], [9]). Furthermore, several PASCAL Challenges have been organised on the unsupervised morphology induction problem in years 2005-2009 (see, e.g., [12]). The evaluations, which now include information retrieval and machine translation applications, have gained a significant interest from the language applications research community. The online demonstration and software packages implementing Morfessor are available at http://www.cis.hut.fi/projects/morpho/ .

4. REFERENCES

[1] K. Koskenniemi, *Two-level morphology: A general computational model for word-form recognition and production*, Ph.D. thesis, Univ. of Helsinki, 1983.

[2] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, vol. 15, World Scientific Series in Computer Science, Singapore, 1989.

[3] J. Goldsmith, "Unsupervised learning of the morphology of a natural language," *Computational Linguistics*, vol. 27, no. 2, pp. 153–198, 2001.

[4] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proc. Workshop on Morphological and Phonological Learning of ACL'02*, Philadelphia, PA, USA, 2002, pp. 21–30.

[5] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Transactions on Speech and Language Processing*, vol. 4, no. 1, Jan. 2007.

[6] P. Grünwald, "A tutorial introduction to the minimum description length principle," in *Advances in Minimum Description Length: Theory and Applications*. 2005, MIT Press.

[7] M. Creutz and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," TKK TR-A81, 2005.

[8] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM TSLP*, vol. 5, no. 1, Dec. 2007.

[9] T. Hirsimäki, J. Pylkkönen, and M. Kurimo, "Importance of high-order n-gram models in morph-based speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 4, pp. 724–732, May 2009.

[10] O. Kohonen, S. Virpioja, and M. Klami, "Allomorfessor: Towards unsupervised morpheme analysis," in *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the CLEF*. 2009, LNCS, Springer-Verlag, To appear.

[11] K. Lagus, O. Kohonen, and S. Virpioja, "Towards unsupervised learning of constructions from text," in *Proc. Workshop on Extracting and Using Constructions in NLP of NODALIDA*, M. Sahlgren and O. Knutsson, Eds., May 2009, SICS Tec. Rep., T2009:10.

[12] M. Kurimo, M. Creutz, and V. Turunen, "Unsupervised morpheme analysis evaluation by ir experiments - morpho challenge 2007," in *Working Notes for the CLEF 2007 Workshop*, A. Nardi and C. Peters, Eds. Sept. 2007, CLEF, Invited paper.