Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology

Stig-Arne Grönroos[1]
stig-arne.gronroos@aalto.fi

Sami Virpioja[2]
sami.virpioja@aalto.fi

Peter Smit[1]
peter.smit@aalto.fi

Mikko Kurimo[1]
mikko.kurimo@aalto.fi

[1]Department of Signal Processing and Acoustics, Aalto University
[2]Department of Information and Computer Science, Aalto University

Abstract

Morfessor is a family of methods for learning morphological segmentations of words based on unannotated data. We introduce a new variant of Morfessor, FlatCat, that applies a hidden Markov model structure. It builds on previous work on Morfessor, sharing model components with the popular Morfessor Baseline and Categories-MAP variants. Our experiments show that while unsupervised FlatCat does not reach the accuracy of Categories-MAP, with semi-supervised learning it provides state-of-the-art results in the Morpho Challenge 2010 tasks for English, Finnish, and Turkish.

1 Introduction

Morphological analysis is essential for automatic processing of compounding and highly-inflecting languages, for which the number of unique word forms may be very large. Apart from rule-based analyzers, the task has been approached by machine learning methodology. Especially unsupervised methods that require no linguistic resources have been studied widely (Hammarström and Borin, 2011). Typically these methods focus on morphological segmentation, i.e., finding *morphs*, the surface forms of the morphemes.

For language processing applications, unsupervised learning of morphology can provide decent-quality analyses without resources produced by human experts. However, while morphological analyzers and large annotated corpora may be expensive to obtain, a small amount of linguistic expertise is more easily available. A well-informed native speaker of a language can often identify the different prefixes, stems, and suffixes of words. Then the question is how many annotated words makes a difference. One answer was provided by Kohonen et al. (2010), who showed that already one hundred manually segmented words provide significant improvements to the quality of the output when comparing to a linguistic gold standard.

The semi-supervised approach by Kohonen et al. (2010) was based on Morfessor Baseline, the simplest of the Morfessor methods by Creutz and Lagus (2002; 2007). The statistical model of Morfessor Baseline is simply a categorical distribution of morphs—a unigram model in the terms of statistical language modeling. As the semi-supervised Morfessor Baseline outperformed all unsupervised and semi-supervised methods evaluated in the Morpho Challenge competitions (Kurimo et al., 2010a) so far, the next question is how the approach works for more complex models.

Another popular variant of Morfessor, Categories-MAP (CatMAP) (Creutz and Lagus, 2005), models word formation using a hidden Markov model (HMM). The context-sensitivity of the model improves the precision of the segmentation. For example, it can prevent splitting a single *s*, a common English suffix, from the beginning of a word. Moreover, it can disambiguate between identical morphs that are actually surface forms of different morphemes. Finally, separation of stems and affixes in the output makes it simple to use the method as a stemmer.

In contrast to Morfessor Baseline, the lexicon of CatMAP is *hierarchical*: a morph that is already in the lexicon may be used to encode the forms of other morphs. This has both advantages and drawbacks.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,
pages 1177–1185, Dublin, Ireland, August 23-29 2014.

One downside is that it mixes the prior and likelihood components of the cost function, so that the semi-supervised approach presented by Kohonen et al. (2010) is not usable.

1.1 Hierarchical versus flat lexicons

From the viewpoint of data compression and following the two-part Minimum Description Length principle (Rissanen, 1978), Morfessor tries to minimize the number of bits needed to encode both the model parameters and the training data. Equivalently, the cost function L can be derived from the Maximum a Posteriori (MAP) estimate:

$$\hat{\theta} = \arg\max_{\theta} \mathrm{P}(\theta \,|\, \boldsymbol{D}) = \arg\min_{\theta} \big(-\log \mathrm{P}(\theta) - \log \mathrm{P}(\boldsymbol{D} \,|\, \theta) \big) = \arg\min_{\theta} L(\theta, \boldsymbol{D}), \tag{1}$$

where θ are the model parameters, $\boldsymbol{D}$ is the training corpus, $\mathrm{P}(\theta)$ is the prior of the parameters and $\mathrm{P}(\boldsymbol{D} \,|\, \theta)$ is the data likelihood.

In context-independent models such as Morfessor Baseline, the parameters include only the forms and probabilities of the morphs in the lexicon of the model. Morfessor Baseline and Categories-ML (CatML) (Creutz and Lagus, 2004) use a flat lexicon, in which the forms of the morphs are encoded directly as strings: each letter requires a certain number of bits to encode. Thus longer morphs are more expensive. Encoding a long morph is worthwhile only if the morph is referred to frequently enough from the words in the training data. If a certain string, let us say *segmentation*, is common enough in the training data, it is cost-effective to have it as a whole in the lexicon. Splitting it into two items, *segment* and *ation*, would double the number of pointers from the data, even if those morphs were already in the lexicon. The undersegmentation of frequent words becomes evident especially if the training data is a corpus instead of a list of unique word forms.

In contrast, Morfessor CatMAP applies a hierarchical lexicon, which makes use of the morphs that are already in the lexicon. Instead of encoding the form of *segmentation* by its 12 letters, we could just encode the form with two references to the forms of the morphs *segment* and *ation*. This may also cause errors, for example encoding *station* with *st* and *ation*.

The lexicon of Morfessor CatMAP allows but does not force hierarchical encoding for the forms: each morph has an extra parameter that indicates whether it has a hierarchical representation or not. The problem of oversegmentation, as in *st* + *ation*, is solved using the morph categories. The categories, which are states of the HMM, include stem, prefix, suffix, and a special non-morpheme category. The non-morpheme category is intended to catch segments that do not fit well into the three proper morph categories because they are fragments of a larger morph. In our example, the morph *st* cannot be a suffix as it starts the word, it is unlikely to be a prefix as it directly precedes a common suffix *ation*, and it is unlikely to be a stem as it is very short. Thus the algorithm is likely to use the non-morpheme state. The hierarchy is expanded only up to the level in which there are no non-morphemes, so the final analysis is still *station*. Without the hierarchy, the non-morphemes have to be removed heuristically, as in CatML (Creutz and Lagus, 2004).

A hierarchical lexicon presents some challenges to model training. For a standard unigram or HMM model, if you know the state and emission sequence of the training data, you can directly derive the maximum likelihood (ML) parameters of the model: a probability of a morph is proportional to the number of times it is referred to, conditional on the state in the HMM. But if the lexicon is partly hierarchical, also the references *within* the lexicon add to the reference counts, and there is no direct way to find the ML parameters even if the encoding of the training data is known. Similarly, semi-supervised learning cannot be accomplished simply by adding the counts from an annotated data set, as it is not clear when to use hierarchy instead of segmenting a word directly in the data.

Moreover, for a flat lexicon, the cost function divides into two parts that have opposing optima: the cost of the data (likelihood) is optimal when there is minimal splitting and the lexicon consists of the words in the training data, whereas the cost of the model (prior) is optimal when the lexicon is minimal and consists only of the letters. In consequence, the balance of precision and recall of the segmentation boundaries can be directly controlled by setting a weight for the data likelihood. Tuning this hyper-parameter is a very simple form of supervision, but it has drastic effects on the segmentation results

(Kohonen et al., 2010). A direct control of the balance may also be useful for some applications: Virpioja et al. (2011) found that the performance of the segmentation algorithms in machine translation correlates more with the precision than the recall. The weighting approach does not work for hierarchical lexicons, for which changing the weight does not directly affect the decision whether to encode the morph with hierarchy or not.

1.2 Morfessor FlatCat

In this paper, we introduce a new member to the Morfessor family, Morfessor FlatCat. As indicated by its name, FlatCat uses a flat lexicon. Our hypothesis is that enabling semi-supervised learning is effective in compensating for the undersegmentation caused by the lack of hierarchy. In particular, semi-supervised learning can improve modeling of suffixation. In the examined languages, suffixes tend to serve syntactic purposes, such as marking case, tense, person or number. Examples are the suffix *s* marking tense and person in *she writes* and number in *stations*. Thus the suffix class is closed and has only a small number of morphemes compared to the prefix and stem categories. As a consequence, a large coverage of suffixes can be achieved already with a relatively small annotated data set.

The basic model of morphotactics in FlatCat is the same as in the CatML and CatMAP variants: a hidden Markov model with states that correspond to a word boundary and four morph categories: stem, prefix, suffix, and non-morpheme. As in CatML, we apply heuristics for removal of non-morphemes from the final segmentation. However, because FlatCat uses MAP estimation of the parameters, these heuristics are not necessary during the training for controlling the model complexity, but merely used as a post-processing step to get meaningful categories.

Modeling of morphotactics improves the segmentation of compound words, by allowing the overall level of segmentation to be increased without increasing the number of correct morphs used in incorrect positions. As the benefits of semi-supervised learning and improved morphotactics are likely to complement each other, we can expect improved performance over the semi-supervised Morfessor Baseline method. By experimental comparison to the previous Morfessor variants, we are able to shed more light on the effects of using an HMM versus unigram model for morphotactics, using a hierarchical versus flat lexicon, and exploiting small amounts of annotated training data.

2 FlatCat model and algorithms

Morfessor FlatCat uses components from the older Morfessor variants. Instead of going through all the details, we refer to the previous work and highlight only the differences. Common components between Morfessor methods are summarized in Table 1.

As a generative model, Morfessor FlatCat describes the joint distribution $\mathrm{P}(\boldsymbol{A}, \boldsymbol{W} \,|\, \theta)$ of words and their analyses. The words $\boldsymbol{W}$ are observed, but their analyses, $\boldsymbol{A}$, is a latent variable in the model. An analysis of a word contains its morphs and morph categories: prefix, stem, suffix, and non-morpheme.

As marginalizing over all possible analyses is generally infeasible, point estimates are used during the training. The likelihood conditioned on the current analyses is

$$\mathrm{P}(\boldsymbol{D} \,|\, \boldsymbol{A}, \theta) = \prod_{j=1}^{|\boldsymbol{D}|} \mathrm{P}(\boldsymbol{A}_j \,|\, \theta). \tag{2}$$

If m_i are the morphs in $\boldsymbol{A}_j$, c_i are the hidden states of the HMM corresponding to the categories of the morphs, and $\#$ is the word boundary, $\mathrm{P}(\boldsymbol{A}_j \,|\, \theta)$ is

$$\mathrm{P}(c_1 \,|\, \#) \prod_{i=1}^{|\boldsymbol{A}_j|} \left[\mathrm{P}(m_i \,|\, c_i)\, \mathrm{P}(c_{i+1} \,|\, c_i) \right] \mathrm{P}(\# \,|\, c_{|\boldsymbol{A}_j|}). \tag{3}$$

Morfessor FlatCat applies an MDL-derived prior designed to control the number of non-zero parameters. The prior is otherwise the same as in Morfessor Baseline, but it includes the usage properties from Morfessor CatMAP: the length of the morph and its right and left perplexity. The perplexity measures describe the predictability of the contexts in which the morph occurs. The emission probability of

	Morfessor method			
Component	Baseline	CatMAP	CatML	**FlatCat**
Lexicon type	Flat	Hierarchy	Flat	Flat
Morphotactics	Unigram	HMM	HMM	HMM
Estimation	MAP	MAP	ML	MAP
Semi-supervised	Implemented	Not implemented	Not implemented	Implemented

Table 1: Overview of similarities and differences between Morfessor methods.

a morph conditioned on the morph category, $\mathrm{P}(m \mid c)$, is calculated from the properties of the morphs similarly as in CatMAP.

2.1 Training algorithms

The parameters are optimized using a local search. Only a part of the parameters are optimized in each step: the parameters that are used in calculating the likelihood of a certain part, *unit*, of the corpus. Units vary in complexity, from all occurrences of a certain morph to the occurrences of a morph bigram whose context fits to certain criteria.

The algorithm tries to simultaneously find the optimal segmentation for the unit and the optimal parameters consistent with that segmentation:

$$(\boldsymbol{A}, \theta) = \underset{\textsc{Op}(\boldsymbol{A},\theta)}{\arg\min} \big\{ L(\theta, \boldsymbol{A}, \boldsymbol{D}) \big\}. \tag{4}$$

The training operators Op define the units changed by the local search and the alternative segmentations tried for each unit. There are three training operators: *split, join* and *resegment*, analogous to the similarly named stages in CatMAP.

The split operator is applied first. It targets all occurrences of a specific morph in the corpus simultaneously, attempting to split it into two parts. The whole corpus is processed by sorting the current morphs by length from shortest to longest.

The second operator attempts to join morph bigrams, grouped by the position of the bigram in the word. The position grouped bigram counts are sorted by frequency, from most to least common.

Finally, resegmenting uses the generalized Viterbi algorithm to find the currently optimal segmentation for one whole word at a time. This operator targets each corpus word in increasing order of frequency.

The heuristics used in FlatCat to remove non-morphemes from the final segmentation are the following: All consequent non-morphemes are joined together. If the resulting morph is longer than 4 characters, it is accepted as a stem. All non-morphemes preceded by a suffix and followed by only suffixes or other non-morphemes are recategorized as suffixes without joining with their neighbors. If any short non-morphemes remain, they are joined either to the preceding or following morphs (the latter only for those in the initial position).

2.2 Semi-supervised learning

Kohonen et al. (2010) found that semi-supervised learning of Morfessor models was not effective by only fixing the values of the analysis $\boldsymbol{A}$ for the annotated samples $\boldsymbol{D}_A$. Their solution was to introduce corpus likelihood weights α and β, one for the unannotated data set and one for the annotated data set. Thus, instead of optimizing the MAP estimate, Kohonen et al. (2010) minimize the cost

$$L(\theta, \boldsymbol{A}, \boldsymbol{D}, \boldsymbol{D}_A) = -\log \mathrm{P}(\theta) - \alpha \log \mathrm{P}(\boldsymbol{D} \mid \boldsymbol{A}, \theta) - \beta \log \mathrm{P}(\boldsymbol{D}_A \mid \boldsymbol{A}, \theta). \tag{5}$$

The weights can be tuned on a development set. We use the same scheme for FlatCat.

The likelihood of the annotated data is calculated using the same HMM that is used for the unannotated data. The morph properties are estimated only from the unannotated data. To ensure that the morphs required for the annotated data can be emitted, a copy of each word in the annotations is added to the

(a) English.

Method	α	β	Pre	Rec	F
U Baseline	1.0	–	.88	.59	.71
U CatMAP	–	–	.89	.51	.65
U FlatCat	1.0	–	.90	.57	.69
W Baseline	0.7	–	.83	.62	.71
W FlatCat	0.5	–	.84	.60	.70
SS Baseline	1.0	3000	.83	**.77**	.80
SS FlatCat	0.9	2000	.86	.76	.81
SS CRF+FlatCat	0.9	2000	.87	**.77**	**.82**
S CRF	–	–	**.92**	.73	.81

(b) Finnish.

Method	α	β	Pre	Rec	F
U Baseline	1.0	–	.84	.38	.53
U CatMAP	–	–	.76	.51	.61
U FlatCat	1.0	–	.84	.38	.52
W Baseline	.02	–	.62	.54	.58
W FlatCat	.015	–	.66	.52	.58
SS Baseline	.1	15000	.75	.72	.73
SS FlatCat	.2	1500	.79	.71	.75
SS CRF+FlatCat	.2	2500	.82	**.76**	.79
S CRF	–	–	**.88**	.74	**.80**

Table 2: Boundary Precision and Recall results in comparison to gold standard segmentation. Abbreviations have been used for Unsupervised (U), likelihood weighted (W), semi-supervised (SS) and fully supervised (S) methods. Best results for each measure have been hilighted using boldface.

unannotated data. This unannotated copy is loosely linked to the annotated word: operations that would result in the removal of a morph required for the annotations from the lexicon cannot be selected, as such an operation would have infinite cost.

3 Experiments

We compare Morfessor FlatCat[1] to two previous Morfessor methods and a fully supervised discriminative segmentation method. The Morfessor methods used as references are the CatMAP[2] and Baseline[3] implementations by Creutz and Lagus (2005) and Virpioja et al. (2013), respectively. Virpioja et al. (2013) implements the semi-supervised method described by Kohonen et al. (2010). For a supervised discriminative model, we use a character-level conditional random field (CRF) implementation by Ruokolainen et al. (2013)[4].

We use the English, Finnish and Turkish data sets from Morpho Challenge 2010 (Kurimo et al., 2010b). They include large unannotated word lists, one thousand annotated words for training, 700–800 annotated words for parameter tuning, and 10 × 1000 annotated words for testing.

For evalution, we use the BPR score by Virpioja et al. (2011). The score calculates the precision (Pre), recall (Rec), and F_1-score (F) of the predicted morph boundaries compared to a linguistic gold standard. In the presence of alternative gold standard analyses, we weight each alternative equally.

We also report the mean average precision from the English and Finnish information retrieval (IR) tasks of the Morpho Challenge. The Lemur Toolkit (Ogilvie and Callan, 2001) with Okapi BM25 ranking was used. The Finnish data consists of 55K documents, 50 test queries and 23K binary relevance assessments. The English data consists of 170K documents, 50 test queries and 20K binary relevance assessments. The domain of both data sets is short newspaper articles. All word forms in both the corpora and the queries were replaced by the morphological segmentation to be evaluated.

Morfessor FlatCat is a pipeline method that refines an initial segmentation given as input. We try two different initializations for the semi-supervised setting: initializing with the segmentation produced by semi-supervised Morfessor Baseline, and initializing with the CRF segmentation. All unsupervised and likelihood-weighted results are initialized with the corresponding Baseline output.

All methods were trained using word types. The weight and perplexity threshold parameters were optimized separately for each method, using a grid search with the held-out data set. The supervised CRF method was trained using the one thousand word annotated training data set.

[1] Available at `https://github.com/aalto-speech/flatcat`

[2] Available at `http://www.cis.hut.fi/projects/morpho/morfessorcatmap.shtml`

[3] Available at `https://github.com/aalto-speech/morfessor`

[4] Available at `http://users.ics.aalto.fi/tpruokol/`

Method	α	β	Pre	Rec	F
U Baseline	1.0	–	.85	.36	.51
U CatMAP	–	–	.83	.50	.62
U FlatCat	1.0	–	.87	.36	.51
W Baseline	0.1	–	.71	.41	.52
W FlatCat	0.3	–	.88	.38	.53
SS Baseline	0.4	2000	.86	.60	.71
SS FlatCat	0.8	2666	.87	.59	.70
SS CRF+FlatCat	1.0	3000	.87	**.61**	**.72**
S CRF	–	–	**.89**	.58	.70

Table 3: Boundary Precision and Recall results in comparison to gold standard segmentation for Turkish. Abbreviations have been used for Unsupervised (U), likelihood weighted (W), semi-supervised (SS) and fully supervised (S) methods. Best results for each measure have been hilighted using boldface.

3.1 Comparison to linguistic gold standards

The results of the BPR evaluations are shown in Tables 2 (English, Finnish) and 3 (Turkish). Semi-supervised FlatCat initialized using CRF achieves the highest F-score for both the English and Turkish data sets. The difference between the highest and second-highest scoring methods is statistically significant for Finnish and Turkish, but not for English (Wilcoxon signed-rank test, $p < 0.01$).

Table 4 shows BPR for subsets of words consisting of different morph category patterns. Each subset consists of 500 words from the English or Finnish gold standard, with one of five selected morph patterns as the only valid analysis. The subsets consist of words with the following morph patterns: words that should not be segmented (STM), compound words consisting of exactly two stems (STM + STM), a prefix followed by a stem (PRE + STM), a stem followed by a single suffix (STM + SUF) and a stem and exactly two suffixes (STM + SUF + SUF). For the STM pattern only precision is reported, as recall is not defined for an empty set of true boundaries.

The fact that semi-supervised FlatCat compares well against CatMAP in recall, for all morph patterns and for the test set as a whole, indicates that supervision indeed is effective in compensating for the undersegmentation caused by the lack of hierarchy in the lexicon. The benefit of modeling morphotactics can be seen in improved precision for the STM + STM (for English and Finnish) and PRE + STM (for Finnish) patterns when comparing against semi-supervised Baseline. The more aggressive segmentation of Baseline gives better results for the English PRE + STM subset than for Finnish due to the shortness of the English prefixes (on average 3.6 letters for the English and 5.3 for the Finnish subset). While not directly observable in Table 4, a large part of the improvement over semi-supervised Baseline is explained by that FlatCat does not use suffix-like morphs in incorrect positions.

Initializing the FlatCat model with CRF segmentation improves the F-scores in all subsets compared to the initialization with Morfessor Baseline. While FlatCat cannot keep the accuracy of the suffix boundaries at as high level as CRF, it clearly improves the stem splitting.

3.2 Information retrieval

Stemming has been shown to improve IR results (Kurimo et al., 2009), by removing inflection that is often not relevant to the query. The morph categories make it possible to simulate stemming by removing morphs categorized as prefixes or suffixes. As longer affixes are more likely to be meaningful, we limited the affix removal to morphs of at most 3 letters. For methods that use morph categories, we report two IR results: the first using all the data and the second with short affix removal (SAR) applied.

In the IR results, we include the topline methods from Morpho Challenge: Snowball Porter stemmer (Porter, 1980) for English and "TWOL first" for Finnish. The latter selects the lemma from the first of the possible analyses given by the morphological analyzer FINTWOL (Lingsoft, Inc.) based on the

(a) English.

	STM	STM + STM			PRE + STM			STM + SUF			STM + SUF + SUF		
Method	Pre	Pre	Rec	F	Pre	Rec	F	Pre	Rec	F	Pre	Rec	F
U CatMAP	**.90**	.94	.63	.75	**.91**	.64	.75	.87	.45	.59	.90	.51	.65
SS Baseline	.64	.93	.77	.84	.82	**.74**	**.77**	.83	.86	.84	.91	.79	.85
SS FlatCat	.68	.94	.65	.77	.78	.62	.69	.86	.88	.87	.94	.79	.86
SS CRF+FlatCat	.68	**.95**	**.78**	**.86**	.78	.66	.72	.87	.89	.88	.94	.80	.87
S CRF	.78	.94	.72	.81	.85	.59	.69	**.92**	**.91**	**.91**	**.95**	**.82**	**.88**

(b) Finnish.

	STM	STM + STM			PRE + STM			STM + SUF			STM + SUF + SUF		
Method	Pre	Pre	Rec	F	Pre	Rec	F	Pre	Rec	F	Pre	Rec	F
U CatMAP	**.77**	.90	**.97**	**.94**	.88	**.96**	**.92**	.67	.46	.54	.68	.38	.49
SS Baseline	.50	.82	.88	.85	.73	.83	.78	.64	.85	.73	.76	.78	.77
SS FlatCat	.49	**.91**	.95	.93	.80	.89	.85	.67	.84	.75	.77	.75	.76
SS CRF+FlatCat	.53	**.91**	.96	**.94**	.84	.94	.88	.71	.88	.79	.80	.79	.79
S CRF	.68	.88	.91	.89	**.90**	.91	.91	**.83**	**.91**	**.87**	**.91**	**.85**	**.88**

Table 4: Results of BPR experiments with different morph category patterns. Best results for each measure have been hilighted using boldface.

two-level model by Koskenniemi (1983). As baseline results we also include unsegmented word forms and truncating each word after the first five letters (First 5).

The results of the IR experiment are shown in Table 5. FlatCat provides the highest score for Finnish. The English scores are similar to those of the semi-supervised Baseline. FlatCat performs better than CRF for both languages. This is explained by the higher level of consistency in the segmentations produced by FlatCat, which makes the resulting morphs more useful as query terms. The number of morphs in the lexicons of FlatCat initialized using CRF are 108 391 (English), 46 123 (Finnish) and 74 193 (Turkish), which is much smaller than the respective morph lexicon sizes counted from the CRF segmentation: 339 682 (English), 396 869 (Finnish) and 182 356 (Turkish). This decrease in lexicon size indicates a more structured segmentation.

The IR performance of semi-supervised FlatCat benefits from the removal of short affixes for English when initialized by CRF, and Finnish for both initializations. It also improves the results of unsupervised FlatCat and CatMAP for Finnish, but lowers the precision for English. A possible explanation is that the unsupervised methods do not analyze the suffixes with a high enough accuracy.

4 Conclusions

We have introduced a new variant of the Morfessor method, Morfessor FlatCat. It predicts both morphs and their categories based on unannotated data, but also annotated training data can be provided. It was shown to outperform earlier Morfessor methods in the semi-supervised learning task for English, Finnish and Turkish.

The purely supervised CRF-based segmentation method proposed by Ruokolainen et al. (2013) outperforms FlatCat for Finnish and reaches the same level for English. However, we show that a discriminative model such as CRF gives inconsistent segmentations that do not work as well in a practical application: In English and Finnish information retrieval tasks, FlatCat clearly outperformed the CRF-based segmentation.

We see two major directions for future work. Currently Morfessor FlatCat, like most Morfessor methods, assumes that words in a sentence occur independently. Making use of the sentence context in which words occur would, however, allow making Part-Of-Speech -like distinctions. These distinctions could

(a) English.

Rank		Method	SAR	MAP
1	–	Snowball Porter	–	0.4092
2	SS	Baseline	–	0.3855
3	**SS**	**FlatCat**	**No**	**0.3837**
4	**SS**	**FlatCat**	**Yes**	**0.3821**
5	**SS**	**CRF+FlatCat**	**Yes**	**0.3810**
6	**SS**	**CRF+FlatCat**	**No**	**0.3788**
7	S	CRF	–	0.3771
8	W	Baseline	–	0.3761
9	U	Baseline	–	0.3695
10	U	CatMAP	No	0.3682
11	U	CatMAP	Yes	0.3653
12	**W**	**FlatCat**	**No**	**0.3651**
13	–	(First 5)	–	0.3648
14	**W**	**FlatCat**	**Yes**	**0.3606**
15	**U**	**FlatCat**	**No**	**0.3486**
16	**U**	**FlatCat**	**Yes**	**0.3451**
17	–	(Words)	–	0.3303

(b) Finnish.

Rank		Method	SAR	MAP
1	**W**	**FlatCat**	**No**	**0.5057**
2	**W**	**FlatCat**	**Yes**	**0.5029**
3	**SS**	**FlatCat**	**Yes**	**0.4987**
4	–	TWOL first	–	0.4973
5	**SS**	**CRF+FlatCat**	**Yes**	**0.4912**
6	U	CatMAP	Yes	0.4884
7	U	CatMAP	No	0.4865
8	**SS**	**CRF+FlatCat**	**No**	**0.4826**
9	**SS**	**FlatCat**	**No**	**0.4821**
10	–	(First 5)	–	0.4757
11	SS	Baseline	–	0.4722
12	S	CRF	–	0.4660
13	W	Baseline	–	0.4582
14	U	Baseline	–	0.4378
15	**U**	**FlatCat**	**Yes**	**0.4349**
16	**U**	**FlatCat**	**No**	**0.4334**
17	–	(Words)	–	0.3483

Table 5: Information Retrieval results. Results of the method presented in this paper are hilighted using boldface. Mean Average Precision is abbreviated as MAP. Short affix removal is abbreviated as SAR.

help disambiguate inflections of different lexemes that have the same surface form but should be analyzed differently (Can and Manandhar, 2013).

The second direction is removal of the assumption that a morphology consists only of concatenative processes. Introducing transformations to model allomorphy in a similar manner as Kohonen et al. (2009) would allow finding the shared abstract morphemes underlying different allomorphs. This could be especially beneficial in information retrieval and machine translation applications.

Acknowledgments

This research has been supported by European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement n°287678 and the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant n°251170) and the LASTU Programme (grants n°256887 and 259934). The experiments were performed using computer resources within the Aalto University School of Science "Science-IT" project. We thank Teemu Ruokolainen for his help with the experiments.

References

Burcu Can and Suresh Manandhar. 2013. Dirichlet processes for joint learning of morphology and PoS tags. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 1087–1091, Nagoya, Japan, October.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In Mike Maxwell, editor, *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30, Philadelphia, PA, USA, July. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2004. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 43–51, Barcelona, Spain, July. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In Timo Honkela, Ville Könönen, Matti Pöllä, and Olli Simula, editors, *Proceedings of AKRR'05,* *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 106–113, Espoo, Finland, June. Helsinki University of Technology, Laboratory of Computer and Information Science.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3:1–3:34, January.

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

Oskar Kohonen, Sami Virpioja, and Mikaela Klami. 2009. Allomorfessor: Towards unsupervised morpheme analysis. In *Evaluating Systems for Multilingual and Multimodal Information Access: 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17–19, 2008, Revised Selected Papers*, volume 5706 of *Lecture Notes in Computer Science*, pages 975–982. Springer Berlin / Heidelberg, September.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden, July. Association for Computational Linguistics.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and production*. Ph.D. thesis, University of Helsinki.

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

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010a. Morpho Challenge 2005-2010: Evaluations and results. In Jeffrey Heinz, Lynne Cahill, and Richard Wicentowski, editors, *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden, July. Association for Computational Linguistics.

Mikko Kurimo, Sami Virpioja, and Ville T. Turunen. 2010b. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo, Finland, September. Aalto University School of Science and Technology, Department of Information and Computer Science. Technical Report TKK-ICS-R37.

Paul Ogilvie and James P Callan. 2001. Experiments using the Lemur toolkit. In *TREC*, volume 10, pages 103–108.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14:465–471.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, Sofia, Bulgaria, August. Association for Computational Linguistics.

Sami Virpioja, Ville T Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University.