Low-Resource Active Learning of North Sámi Morphological Segmentation

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

Kristiina Jokinen[2]
kristiina.jokinen@helsinki.fi

Katri Hiovain[2]
katri.hiovain@helsinki.fi

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

Sami Virpioja[3]
sami.virpioja@aalto.fi

[1]Department of Signal Processing and Acoustics, Aalto University, Finland
[2]Institute of Behavioural Sciences, University of Helsinki, Finland
[3]Department of Information and Computer Science, Aalto University, Finland

December 15, 2014

Abstract

Many Uralic languages have a rich morphological structure, but lack tools of morphological analysis needed for efficient language processing. While creating a high-quality morphological analyzer requires a significant amount of expert labor, data-driven approaches may provide sufficient quality for many applications. We study how to create a statistical model for morphological segmentation of North Sámi language with a large unannotated corpus and a small amount of human-annotated word forms selected using an active learning approach. For statistical learning, we use the semi-supervised Morfessor Baseline and FlatCat methods. After annotating 237 words with our active learning setup, we improve morph boundary recall over 20% with no loss of precision.

1 Introduction

In morphologically rich languages, such as the Uralic languages, the number of observed word forms grows rapidly with increasing corpus size. This vocabulary growth can be problematic for natural language processing (NLP) applications, because it causes sparsity in the calculated statistics. Thus it is essential to model such languages on a sub-word level, using for example morphological analysis.

Despite the improvement of development tools and increase of computational resources since the introduction of finite-state transducer (FST) based morphological analyzers in the 1980s [1], the bottleneck for the traditional method of building such analyzers is still the large amounts of manual labor and skill that are required [2]. The strength of such analyzers is the potential to produce output of high quality and richly informative morphological tags.

Morphological surface segmentation is a relaxed variant of morphological analysis, in which the surface form of a word is divided into segments that correspond to morphemes. The segments, called *morphs*, are not mapped onto underlying abstract morphemes as in FST-based analyzers, but concatenating the sequence of morphs results directly in the observed word form. Allomorphic variation is left unresolved.

Although unsupervised learning of morphological segmenters does not reach the detail and accuracy of hand-built analyzers, it has proven useful for many NLP applications, including speech recognition [3], information retrieval [4], and machine translation [5]. Unsupervised methods are especially valuable for low-resource languages, as they do not require any expensive resources produced by human experts.

While hand built morphological analyzers and large annotated corpora may be unavailable due to the expense, a small amount of linguistic expertise is easier to obtain. Given word forms embedded in sentence contexts, a well-informed native speaker of a language can mark the prefixes, stems, and suffixes of the words in question. A brief collection effort of this type will result in a very small set of annotated words.

Small annotated data of this type can be used to augment large unannotated data by using semi-supervised methods, which are able to learn from such mixed data. As little as one hundred manually segmented words have been shown to provide significant improvements to the quality of the output when comparing to a linguistic gold standard [6]. Adding more annotated data improves the results, with rapid improvement to one thousand words or beyond.

When gathering annotated training samples for a specific model, *active learning* may provide better results than selecting the samples randomly. In each iteration of active learning, the current best model, trained with all training samples collected up to that point, is used in selection of the new samples to annotate for the next iteration. In this work, we use active learning for morphological segmentation of North Sámi.

1.1 North Sámi

North Sámi (davvisámegiella) belongs to Finno-Ugrian languages and is related to Finnish and other Baltic-Finnic languages. It is one of the nine Sámi languages spoken in the Northern Polar Cap, and spread along Norway, Sweden, Finland and Russia. The speakers of the Sámi languages do not necessarily understand each other but the languages form a chain of adjacent groups. North Sámi is the most widely used Sámi language with around 20 000 speakers, functioning as a lingua franca among the Sámi speakers and used in text books, children's books, newspapers, and broadcasts.

Linguistically North Sámi is characterized as an inflected language, with cases, numbers, persons, tense and mood. The inflectional system has seven cases. It is accompanied by complicated although regular morphophonological variation. The inflected forms follow weak and strong grades which concern almost all consonants. North Sámi is also fusional: a single word form can stand for more than one morphological category. The nouns have four inflection categories (stems with a vowel or a consonant, the so-called contracting is-nouns, and alternating u-nouns), while the verbs have three conjugation categories (gradation, three syllable, two syllable). The only one syllable verbs are "leat" (to be) and the negation verb. In syntax, the Sámi has separate dual forms for pronouns and verbs besides singular and plural forms.

1.2 Related work

While unsupervised morphological segmentation has recently been an active topic of research [7], semi-supervised morphological segmentation has not received as much attention. One approach is to seed the learning with a small amount of linguistic knowledge in addition to the unannotated corpus [8]. Some semi-supervised methods where a part of the training corpus is supplied with correct outputs have also been presented, including generative [6, 9, 10] and discriminative [11, 12] methods.

Active learning methods have been applied for constructing FST-based analyzers by eliciting new rules from a user with linguistic expertise [13, 14]. These development efforts are fast for rule-based systems, but still require months of work. There has been research effort into FST-based morphology for Sámi languages [15, 16, 17]

North Sámi is the focus of the DigiSami project, which attempts to increase the digital viability of minor Finno-Ugric languages by technology development, analysis, data collection (read and conversational speech), and encouragement of community effort in online content creation [18, 19]. This work directly supports the ultimate goal of the project, which is to produce tools and technology that would allow Sámi speech-based applications to be developed. Although North Sámi has various linguistic resources, there are not many related to speech technology.

2 Methods

As a method for morphological segmentation of words, we use Morfessor. It is a family of methods for learning morphological segmentations primarily from unannotated data. The methods are based on a generative probabilistic model which generates the observed word forms by concatenating morphs. The model parameters $\boldsymbol{\theta}$ define a *morph lexicon*. The morph m_i is considered to be stored in the morph lexicon, if it has a non-zero probability $P(m \mid \boldsymbol{\theta})$ given the parameters.

Morfessor utilizes a prior distribution $P(\boldsymbol{\theta})$ over morph lexicons, derived from the Minimum Description Length principle [20]. The prior favors lexicons that contain fewer, shorter morphs. The purpose is to balance the size of the lexicon, and the size of the corpus $\boldsymbol{D}$ when encoded using the lexicon. This balance can be expressed as finding the following Maximum a Posteriori (MAP) estimate:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta} \mid \boldsymbol{D}) = \arg\min_{\boldsymbol{\theta}} \big(-\log P(\boldsymbol{\theta}) - \log P(\boldsymbol{D} \mid \boldsymbol{\theta}) \big). \tag{1}$$

In the Morfessor variants used in this work, the lexicon encodes the forms of the morphs directly as strings: each letter requires a certain number of bits to encode.

2.1 Morfessor Baseline

Morfessor Baseline [21, 22] employs a morph lexicon $P(m \mid \boldsymbol{\theta})$ that is simply a categorical distribution over morphs m, in other words a unigram model. The model parameters $\boldsymbol{\theta}$ are optimized utilizing a greedy local search, in which one training word at a time is reanalyzed and the model parameters updated accordingly.

In order to use the annotations produced in the active learning for training Morfessor, we employ the semi-supervised extension to Morfessor Baseline [6]. This involves replacing the MAP estimate (1) with the optimization

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \big(-\log P(\boldsymbol{\theta}) - \log \alpha P(\boldsymbol{D} \mid \boldsymbol{\theta}) - \log \beta P(\boldsymbol{A} \mid \boldsymbol{\theta}) \big), \tag{2}$$

where $\boldsymbol{D}$ is the unannotated and $\boldsymbol{A}$ the annotated training corpus, α and β are the weights for the likelihood of the unannotated corpus and annotated corpus, respectively. The hyper-parameters α and β affect the overall amount of segmentation and the relative importance of using the morphs present in the annotated corpus.

2.2 Morfessor FlatCat

The most recent Morfessor variant is called Morfessor FlatCat [10]. The main difference between Morfessor Baseline and Morfessor FlatCat is the use of morph categories

in the latter. Each morph token is categorized as PREFIX, STEM, or SUFFIX. Internally to the algorithm, a NON-MORPH category is used, intended to model frequent substrings that are not morphs but fragments of a morph. Word formation is modeled using a hidden Markov model (HMM) having morph categories as hidden states and morphs as observations. HMM morphotactics were previously used in the Categories-ML [23] and Categories-MAP [24] variants of Morfessor, but Morfessor FlatCat is the first method to combine the approach with semi-supervised training.

The benefit of the HMM morphotactics is increased context-sensitivity, which improves the precision of the segmentation. For example, in English, the model can prevent splitting a single *s*, a common suffix, from the beginning of a word. Modeling of morphotactics also improves the segmentation of compound words, by allowing the overall level of segmentation to be increased. The main benefits of semi-supervised learning are in the modeling of suffixation. [10]

The prior of Morfessor FlatCat is otherwise the same as in Morfessor Baseline, but it also includes encoding of the right and left perplexity of the morph. The perplexity measures describe the predictability of the contexts in which the morph occurs. The perplexities, together with the length of the morph, are used to calculate the emission probability of a morph conditioned on the morph category, $P(m \mid c)$.

2.3 Pool-based active learning

Pool-based active learning [25] has been successfully applied in NLP [26]. In pool-based active learning, the system has access to a pool of unlabeled data $\mathcal{A}$ and can request from the annotator true labels for a certain number of samples in the pool.

A method for choosing which samples to annotate still needs to be defined. A well suited approach for generative models is to use the model's estimate of the uncertainty of the decision associated with a particular sample in order to select the additional samples to annotate [27]. In the case of morphological segmentation, we use the uncertainty of a word's current segmentation in order to assess its value as an additional annotation. The next word to annotate $\boldsymbol{A}_{(t+1)}$ at time step t is selected from $\mathcal{A}$ based on the uncertainty of the current best segmentation Z_i

$$\boldsymbol{A}_{(t+1)} = \underset{\mathrm{w}_i \in \mathcal{A}}{\arg\min} \frac{P(Z_i \mid \boldsymbol{\theta}_t)}{P(\mathrm{w}_i \mid \boldsymbol{\theta}_t)}, \tag{3}$$

where the likelihood of the current segmentation $P(Z_i \mid \boldsymbol{\theta}_t)$ is given by the Viterbi algorithm [28] and the likelihood of the word with any segmentation $P(\mathrm{w}_i \mid \boldsymbol{\theta}_t)$ is given by the forward algorithm [29].

Corpus	Word tokens	Word types
Den samiske tekstbanken	17 985 140	691 190
UIT-SME-TTS	42 150	8194
Development set	–	100
Evaluation pool	–	900
Training pool $\mathcal{A}$	–	7194

Table 1: Sizes of the unannotated corpora and the initial division into subsets.

3 Experiments

We used two different text corpora in our experiments. The sizes of the corpora are shown in Table 1. The larger *Den samiske tekstbanken* corpus was only used to construct a word list, to use as the unannotated training data. The smaller *UIT-SME-TTS* corpus was divided into separate pools from which evaluation and training words were drawn for annotation. The sentences in which the words occur were also extracted for use as contexts. To ensure that the evaluation words are unseen, the words in the evaluation pool were removed from the other subsets.

The use of two corpora enables the release of the annotations with their sentence contexts, which would have been precluded by the restrictive corpus of the *Tekstbanken* corpus. It also demonstrates the effectiveness of the system under the realistic scenario where a large general-domain word list for the language is available for use, even though the corpora themselves are restricted by licensing. A similar scenario would be selection from a specific target domain corpus.

Initially we use Morfessor Baseline, but towards the end of the experiment we switch the method to Morfessor FlatCat. As prefixes are very rare in North Sámi, and none were seen in the annotations, we disabled the prefix category.

3.1 Active learning

Our active learning procedure starts from nothing but an unannotated corpus collected for other purposes. An initial model is trained in an unsupervised fashion. The procedure then applies three components iteratively: (i) selection of new words to annotate using the current model, (ii) elicitation of annotations for the selected words, and (iii) training of the new segmentation model using all available training data.

For the elicitation step, we developed a web-based annotation interface. A javascript app using the jQuery framework was used as a front-end and a RESTful Python wsgi-app built on the bottle framework as a back-end. Screenshots of the annotation

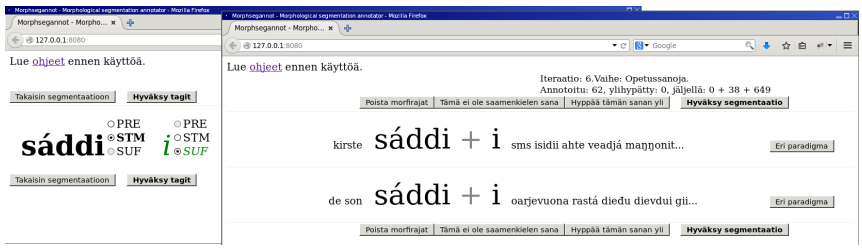

Figure 1: Screenshots of the annotation interface.

interface are shown in Figure 1. For words in the training pool, the interface shows the segmentation of the current model as a suggestion to the annotator. Words in the evaluation pool are shown unsegmented, in order not to bias the annotator.

There are no efficient on-line training algorithms for Morfessor FlatCat. Thus we gather a list of 50 new words to annotate, by ranking the potential words according to (3), and re-train once the whole list has been annotated. Re-training includes hyper-parameter optimization (HPO) for α and β. Due to the very limited amount of training data, and a lack of previously collected annotated development set, we initially decided to use 3-fold cross-validation on the annotated training set for HPO. This initial approach was quickly shown to be flawed, as the values of the hyper-parameters did not begin to converge after multiple iterations. This divergence can be explained by HPO requiring the development set to be an unbiased sample of the data distribution. A subset biased towards maximally informative words is desired for use as training words, but using them for HPO introduces an undesired bias.

To remedy this situation, we constructed a development set of 100 randomly selected words with annotations. We then restarted the iterations, now using the development set for HPO. In this second approach, when a word that had already been annotated was reselected for annotation, the old annotation was used, making it unnecessary to re-elicit from the annotator. The training iterations and the respective hyper-parameter values are shown in Table 2.

3.2 Annotation details

The annotations were produced by a single trained linguist, who is not a native speaker of Sámi. In total 457 randomly selected word types and 346 actively selected word types were annotated under a time span of 17 days. The total time spent by the annotator was 19 hours (over 30 min breaks omitted).

			Hyper-parameters		Annotated words			Test
	Iteration	**Training**	α	β	**Dev**	**Train**	**Tot**	F_1
C	0	U Baseline	0.42	–	–	–	0	.67
C	1	U Baseline	1.3	–	–	–	49	.68
C	2	S Baseline	2.4	1300	–	98	98	.67
C	3	S Baseline	2.7	800	–	148	148	.66
C	4	S Baseline	3.4	900	–	198	198	.66
D	0	U Baseline	1.1	–	100	–	100	.68
D	1	S Baseline	1.4	800	100	50	150	.69
D	2	S Baseline	1.5	700	100	123	223	.70
D	3	S FlatCat	0.2	1400	100	182	282	.74
D	4	S FlatCat	0.5	2200	100	237	337	.76

Table 2: The model parameters and number of annotations for the active learning iterations. U and S stand for unsupervised and semi-supervised training, C and D for setting hyper-parameters by cross-validation and development set, respectively.

Most of the annotated word tokens had an unambiguous segmentation agreeing with established linguistic interpretation. These words contain only easily separated suffixes: markers for case and person, and derivational endings. However, some words required the annotator to make choices on where to place the boundary.

One challenge was posed by the extensive stem alternation and fusion in Sámi. To maximize consistency, the segmentation boundary was usually placed so that all of the morphophonological alternation remains in the stem. Exceptions include the passive derivational suffix, which is found as variants *-ojuvvo-* and *-juvvo-* depending on the inflectional category and stem type. Another challenge were lexicalized stems. These stems appear to end with a derivational suffix, but removal of the suffix does not yield a morpheme at all, or results in a morpheme with very weak semantic relation to the lexicalized stem. An example is *ráhkadit* (make, produce).

3.3 Evaluation

The word segmentations generated by the model are evaluated by comparison with annotated morph boundaries using *boundary precision*, *boundary recall*, and *boundary F_1-score* [30]. The boundary F_1-score equals the harmonic mean of precision (the percentage of correctly assigned boundaries with respect to all assigned boundaries) and recall (the percentage of correctly assigned boundaries with respect to the reference boundaries). Precision and recall are calculated using macro-averages over the

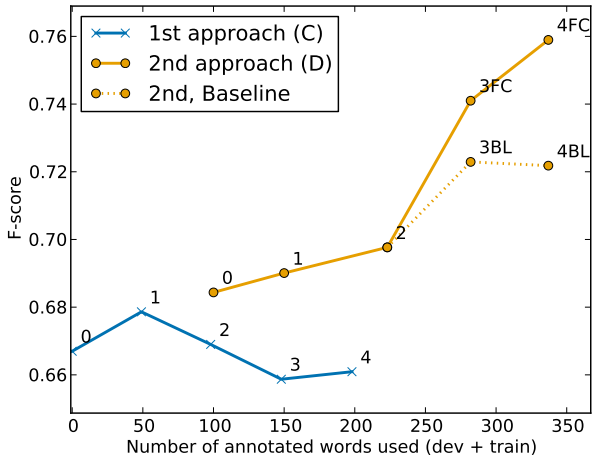

Figure 2: F_1-score for models trained using varying amounts of annotated data. The labels indicate the iteration number. BL stands for Baseline and FC for FlatCat.

words in the evaluation set. In the case that the word has more than one annotated segmentation, we take the one that gives the highest score.

We also report the scores for subsets of words consisting of different morph category patterns found in the evaluation set. These categories are words that should not be segmented (STM), compound words consisting of exactly two stems (STM+STM), a stem followed by a single suffix (STM+SUF) and a stem and exactly two suffixes (STM+SUF+SUF). Only precision is reported for the STM pattern, as recall is not defined for an empty set of true boundaries.

3.4 Results

Figure 2 shows the improvement of the F_1-score as more annotations became available. Training a Morfessor FlatCat model after three iterations provided a large boost, even though the annotated words had so far been selected by Baseline models. In contrast, the words selected by the FlatCat model (3FC) for annotation did not benefit the Baseline model (4BL).

Table 3 shows scores for sets of words with different morphological patterns. For the full test set, we improve morph boundary recall over 20% (relative) with no loss of precision, when comparing the first model of the second approach (D0) to the last

	STM	STM+STM			STM+SUF			STM+SUF+SUF			Full test set		
Model	Pre	Pre	Rec	F_1	Pre	Rec	F_1	Pre	Rec	F_1	Pre	Rec	F_1
C0 Baseline	.57	.85	.88	.87	.63	.46	.53	.65	.33	.44	.66	.67	.67
D0 Baseline	.70	**.89**	.85	.87	.74	.38	.50	**.71**	.31	.43	**.76**	.62	.68
D4 FlatCat	**.73**	.85	**.92**	**.89**	**.75**	**.62**	**.68**	.60	**.38**	**.46**	**.76**	**.75**	**.76**

Table 3: Boundary precision (Pre), recall (Rec), and F_1-scores for different subsets of the evaluation data.

model (D4). The performance has improved for all morph patterns. The STM+SUF pattern has the largest increase, with improvements both in precision and recall. The recall scores of compound words (STM+STM) and suffix sequences (STM+SUF+SUF) are also clearly improved.

4 Conclusions

We have applied an active learning approach to modeling morphological segmentation of North Sámi. The work was accomplished using open-source software [1]. We present the collected language resources for the use of the scientific community [2].

The performance of the segmentation model was shown to increase rapidly as the amount of human-annotated data was increased. One of our findings is the importance of collecting an unbiased development set for optimization of hyper-parameters, even though this reduces the amount of human-labeled data available for training. Cross-validating using the selected samples is not an adequate compromise.

One avenue for future work is exploring other measures for selecting the words to annotate. These can include applying other language models, but can also be based on direct statistics of the language, e.g. frequencies and lengths of the words or substrings of the words. A thorough comparison to random selection should also be performed. Another question is how well this approach extends to other languages and corpora.

Acknowledgments

This research has been supported by EC's Seventh Framework Programme under grant n°287678 and the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant n°251170), LASTU Programme (grants n°256887 and 259934) and Fenno-Ugric Digital Citizens (grant n°270082). Computer resources within the Aalto University School of Science "Science-IT" project were used.

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

[2] Available at `http://research.spa.aalto.fi/speech/data_release/north_saami_active_learning/` .

References

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

[2] Kimmo Koskenniemi. How to build an open source morphological parser now. In *Resourceful Language Technology–Festschrift in Honor of Anna Sågvall Hein*, page 86. 2008.

[3] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1):3:1–3:29, December 2007.

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

[5] Harri Kirik and Mark Fishel. Modelling linguistic phenomena with unsupervised morphology for improving statistical machine translation. In *Proceedings of the SLTC'08 Workshop on Unsupervised Methods in NLP, Stockholm, Sweden*, 2008.

[6] Oskar Kohonen, Sami Virpioja, and Krista Lagus. 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 2010. Association for Computational Linguistics.

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

[8] David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216. Association for Computational Linguistics, 2000.

[9] Kairit Sirts and Sharon Goldwater. Minimally-supervised morphological segmentation using adaptor grammars. *TACL*, 1:255–266, 2013.

[10] Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 1177–1185. ACL, 2014.

[11] Hoifung Poon, Colin Cherry, and Kristina Toutanova. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217. Association for Computational Linguistics, 2009.

[12] Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84–89, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[13] Kemal Oflazer, Sergei Nirenburg, and Marjorie McShane. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational Linguistics*, 27(1):59–85, 2001.

[14] Sonja E Bosch, Laurette Pretorius, Kholisa Podile, and Axel Fleisch. Experimental fast-tracking of morphological analysers for Nguni languages. In *LREC*, 2008.

[15] Trond Trosterud and Heli Uibo. Consonant gradation in Estonian and Sámi: two-level solution. In *Inquiries into Words, Constraints and Contexts–Festschrift for Kimmo Koskenniemi on his 60th Birthday*, page 136. Citeseer, 2005.

[16] Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. Hfst tools for morphology–an efficient open-source package for construction of morphological analyzers. In *State of the Art in Computational Morphology*, pages 28–47. Springer, 2009.

[17] Francis M Tyers, Linda Wiechetek, and Trond Trosterud. Developing prototypes for machine translation between two Sámi languages. In *Proceedings of the 13th Annual Conf. of the EAMT*, pages 120–128, 2009.

[18] Kristiina Jokinen and Graham Wilcock. Multimodal open-domain conversations with the Nao robot. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 213–224. Springer, 2013.

[19] Kristiina Jokinen and Graham Wilcock. Community-based resource building and data collection. In *The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14)*, 2014.

[20] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Series in Computer Science, Singapore, 1989.

[21] Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL'02*, pages 21–30, Philadelphia, Pennsylvania, USA, 2002.

[22] Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1), jan 2007.

[23] Mathias Creutz and Krista Lagus. 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, July 2004.

[24] Mathias Creutz and Krista Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the AKRR'05*, Espoo, Finland, 2005.

[25] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[26] Andrew Kachites McCallumzy and Kamal Nigamy. Employing EM and pool-based active learning for text classification. In *Machine Learning: Proceedings of the Fifteenth International Conference, ICML*. Citeseer, 1998.

[27] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

[28] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[29] Leonard E. Baum. An inequality and an associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3(1):1–8, 1972.

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