

# Weighting Finite-State Morphological Analyzers using HFST Tools \*

Krister Lindén

Tommi Pirinen

University of Helsinki

Helsinki, Finland

{krister.linden,tommi.pirinen}@helsinki.fi

March 6, 2010

## Abstract

In a language with very productive compounding and a rich inflectional system, e.g. Finnish, new words are to a large extent formed by compounding. In order to disambiguate between the possible compound segmentations, a probabilistic strategy has been found effective by Lindén and Pirinen [7]. In this article, we present a method for implementing the probabilistic framework as a separate process which can be combined through composition with a lexical transducer to create a weighted morphological analyzer. To implement the analyzer, we use the HFST-LEXC and related command line tools which are part of the open source *Helsinki Finite-State Technology* package. Using Finnish as a test language, we show how to use the weighted finite-state lexicon for building a simple unigram tagger with 97 % precision for Finnish words and word segments belonging to the vocabulary of the lexicon.

## 1 Introduction

In English the received wisdom is that traditional morphological analysis is too complex for statistical taggers to deal with; a simplified tagging scheme is needed. The disambiguation accuracy will otherwise be too low even with an n-gram tagger because there is not enough training material. However, currently training material for morphological disambiguators is abundantly available. At the same time, one could argue that the interest in tagging has disappeared, because we can do more complex things such as syntactic dependency analysis and get the morphological disambiguation as a side effect. As a matter of curiosity, we will still pursue statistical tagging, because there is also the initial result often attributed to Ken Church that approximately 90 % of the readings in English will be correct if one simply gives each word its most frequent morphosyntactic tag. We wish to derive a similar baseline for Finnish.

In addition, a morphologically complex language like Finnish is different than English. In English there are hardly any inflectional endings and applying

---

\*This is author's draft; it may differ from published version slightly, especially since hyperref doesn't play nicely with llncs

traditional morphological analysis to English necessarily creates massive ambiguity that can only be resolved by context, whereas morphologically complex languages like Finnish in each word most often carry the morphemes referred to by the morphological tags. As the morphological tags have a physical correspondence in the strings, it should be possible to use much less context, or perhaps none at all, to disambiguate the traditional morphological analysis of languages like Finnish. After all, the reduced tag sets of English statistical taggers can be viewed as an attempt to simplify the tag set to refer only to the visible surface morphemes in a locally constrained context.

There are some initial encouraging results by Lindén and Pirinen [7] for disambiguating Finnish compounds using unigram statistics for the parts in a productive compound process. Unigram statistics for compounds is essentially the same as taking the most likely morpheme segmentation and the most frequent reading of each compound word. Similar results for disambiguating compounds using a slightly different basis for estimating the probabilities have been demonstrated for German by Schiller [11] and by Marek [9]. These results further encourage us to pursue the topic of full morphological tagging for a complex language like Finnish using only a lexicon and unigram statistics for the words and their compound parts.

In [7], Lindén and Pirinen suggest a method which essentially requires the building of a full form lexicon and an estimate for each separate word form. This is not particularly convenient, instead we introduce a simplified way to weight the different parts of the lexicon with frequency data from a corpus by using weighted finite-state transducer calculus. We use the open source software tools of HFST<sup>1</sup>, which contains HFST-LEXC similar to the Xerox LexC tool [2]. In addition to compiling LexC-style lexicons, HFST-LEXC has a mechanism for adding weights to compound parts and morphological analyses. The HFST tools also contain a set of command line tools that are convenient for creating the final weighted morphological analyzer using transducer calculus.

We apply the weighted morphological analyzer to the task of morphologically tagging Finnish text. As expected, it turns out that a highly inflecting and compounding language with a free word order like Finnish solves many of its linguistic ambiguities during word formation. This pays back in the form of 97 % tagger precision using only a very simple unigram tagger in the form of a weighted morphological lexicon for the words and word parts that are in the lexicon. For words that contain unknown parts, the lexicalized strategy is, however, rather toothless. For such words it seems, we may, after all, need a traditional guesser and n-gram statistics for morphological disambiguation.

The remainder of the article is structured as follows. In Sections 2, we briefly present some aspects of Finnish morphology that may be problematic for statistical tagging. In Section 3, we introduce the probabilistic formulation of how to weight lexical entries. In Section 4, we introduce the test and training corpora. In Section 5, we evaluate the weighted lexicon on tagging Finnish text. Finally, in Sections 6 and 7, we discuss the results and draw the conclusions.

---

<sup>1</sup>[hfst.sourceforge.net](http://hfst.sourceforge.net)

## 2 Finnish Morphology

We present some aspects of Finnish inflectional and compounding morphology that may be problematic for statistical tagging in Sections 2.1 and 2.2. For a more thorough introduction to Finnish morphology, see Karlsson [5], and for an implementation of computational morphology, see Koskenniemi [6]. In Section 2.2, we present an outline of how to implement the morphology in sublexicons which are useful for weighting.

### 2.1 Inflection in Finnish

In Finnish morphology, the inflection of typical nouns produces several thousands of forms for the productive inflection. E.g. a noun has more than 12 cases in singular and plural as well as possessive suffixes and clitic particles resulting in more than 2000 forms for every noun.

Mostly the traditional linguistically motivated morphological analysis of Finnish is based on visible morphemes. However, for illustrational purposes we will discuss two prototypical cases where the analysis needs context. One such case is where a possessive suffix overrides the case ending to create ambiguity: *taloni* 'my house/of my house/my houses', i.e. either *talo* 'house' nominative singular, *talon* 'of the house' genitive singular or *talot* 'houses' nominative plural followed by a possessive suffix. This ambiguity is systematic, so either the distinctions can be left out or one can create a complex underspecified tag *+Sg+Nom/+Sg+Gen/+Pl+Nom* for this case.

Another case, which is common in most languages, is the distinction between nouns or adjectives and participles of verbs. This often affects the choice of baseform for the word, i.e. the baseform of 'writing' is either a verb such as 'write' or a noun such as 'writing'. In Finnish, we have words like *taitava* 'skillful Adjective' or 'know Verb Present Participle' and *kokenut* 'experienced Adjective' or 'experience Verb Past Participle'. Since the two readings have different baseforms, it is not possible to defer the ambiguity to be resolved later by using underspecification. In some cases, one of the forms is rare and can perhaps be ignored with a minimal loss of information, but sometimes both occur regularly and in overlapping contexts, in which case both forms should be postulated and eventually disambiguated. However, sufficient information for doing this reliably may not be available before some degree of syntactic or semantic analysis.

In Sections 5 and 6, we will return to the significance of these problems in Finnish and their impact on the morphological disambiguation.

### 2.2 Compounding in Finnish

Finnish compounding theoretically allows nominal compounds of arbitrary length to be created from initial parts of certain noun forms. The final part may be inflected in all possible forms.

Normal inflected Finnish noun compounds correspond to prepositional phrases in English, e.g. *ostoskeskuksessa* 'in the shopping center'. The morphological analysis in Finnish of the previous phrase into *ostos#keskus+N+Sg+Ine* corresponds in English to noun chunking and case analysis into 'shopping center +N+Sg+Loc:In'.

In extreme cases, such as the compounds describing ancestors, nouns are compounded from zero or more of *isän* ‘father SINGULAR GENITIVE’ and *äidin* ‘mother SINGULAR GENITIVE’ and then one of the inflected forms of *isä* or *äiti* creating forms such as *äidinisälle* ‘to (maternal) grandfather’ or *isänisänisänisä* ‘great great grandfather’. As for the potential ambiguity, Finnish also has the noun *nisä* ‘udder’, which creates ambiguity for any paternal grandfather, e.g. *isän#isän#isän#isä*, *isän#isä#nisän#isä*, *isä#nisä#nisä#nisä*, ...

Finnish compounding also includes forms of compounding where all parts of the word are inflected in the same form, but this is limited to a small fraction of adjective initial compounds and to the numbers if they are spelled out with letters. In addition, some inflected verb forms may appear as parts of compounds. These are much more rare than nominal compounds [4] so they do not interfere with the regular compounding.

### 2.3 Finnish Computational Morphology

Pirinen [10] presented an open source implementation of a finite state morphological analyzer for Finnish, which has been reimplemented with the HFST tools and extended with data collected and classified by Listenmaa [8]. We use the reimplemented and extended version as our unweighted lexicon. Pirinen’s analyzer has a fully productive noun compounding mechanism. Fully productive noun compounding means that it allows compounds of arbitrary length with any combination of nominative singulars, genitive singulars, or genitive plurals in the initial part and any inflected form of a noun as the final part.

The morphotactic combination of morphemes is achieved by combining sublexicons as defined in [2]. We use the open source software called HFST-LEXC with a similar interface as the Xerox LexC tool. The interested reader is referred to [2] for an exposition of the LexC syntax. The HFST-LEXC tool extends the syntax with support for adding weights on the lexical entries.

We note that the noun compounding can be decomposed into two concatenatable lexicons separated by a word boundary marker, i.e. any number of noun prefixes *CompoundNonFinalNoun*\* in Figure 1 separated by ‘#’ and from the inflected noun forms *CompoundFinalNoun* in Figure 2. Similar decompositions can be achieved for other parts of speech as needed. For a further discussion of the structure of the lexicon, see [7].

```

LEXICON Root
## CompoundNonFinalNoun ;
## #;

LEXICON Compound
#:0 CompoundNonFinalNoun;
#:0 #;

LEXICON CompoundNonFinalNoun
isä  Compound "weight: 0, gloss: father" ;
isän Compound "weight: 0, gloss: father's" ;
äiti Compound "weight: 0, gloss: mother" ;
äidin Compound "weight: 0, gloss: mother's" ;

```

Figure 1: Unweighted fragment for  $\{CompoundNonFinalNoun\}^*$  i.e. *noun prefixes*.

```

LEXICON Root
CompoundFinalNoun ;

LEXICON CompoundFinalNoun
isä:isä+sg+nom    ## "weight: 0, gloss: father" ;
isän:isä+sg+gen   ## "weight: 0, gloss: father's" ;
isälle:isä+sg+all ## "weight: 0, gloss: to the father" ;

LEXICON ##
## # ;

```

Figure 2: Unweighted fragment for *CompoundFinalNoun*, i.e. *noun forms*.

### 3 Methodology

Assume that we want to know the probability of a morphological analysis with a morpheme segmentation  $A$  given the token  $a$ , i.e.  $P(A|a)$ . According to Bayes rule, we get Equation 1.

$$P(A|a) = P(A, a)/P(a) = P(a|A)P(A)/P(a) \quad (1)$$

We wish to retain only the most likely analysis and its segmentation  $A$ . As we know that  $P(a|A)$  is almost always 1, i.e. a word form is known when its analysis is given. Additionally,  $P(a)$  is constant during the maximization, so the expression simplifies to finding the most likely global analysis  $A$  as shown by Equation 2, i.e. we only need to estimate the output language model.

$$\arg \max_A P(A|a) = \arg \max_A P(a|A)P(A)/P(a) = \arg \max_A P(A) \quad (2)$$

In order to find the most likely segmentation of  $A$ , we can make the additional assumption that the probability  $P(A)$  is proportional to the product of the probabilities  $P(s_i)$  of the segments of  $A$ , where  $A = s_1s_2\dots s_n$ , defined by Equation 3. This assumption based on a unigram language model of compounding has been demonstrated by Lindén and Pirinen [7] to work well in practice.

$$P(A) \propto \prod_{s_i} P(s_i) \quad (3)$$

#### 3.1 Estimating probabilities

The estimated probability of a token,  $a$ , to occur in the corpus is proportional to the count,  $c(a)$ , divided by the corpus size,  $cs$ . The probability  $p(a)$  of a token in the corpus is defined by Equation 4. We also note that the corpus estimate for  $p(a)$  is in fact an estimate of the sum of the probabilities of all the possible analyses and segmentations of  $a$  in the corpus.

$$p(a) = c(a)/cs \quad (4)$$

Tokens  $x$  known to the original lexicon but unseen in the corpus need to be assigned a small probability mass different from 0, so they get  $c(x) = 1$ , i.e. we define the count of a token as its corpus frequency plus 1 as in Equation 5, also known as Laplace smoothing.

$$c(a) = 1 + \text{frequency}(a) \quad (5)$$

### 3.2 Weighting the Lexicon

In order to use the probabilities as weights in the lexicon, we implement them in the tropical semiring, which means that we use the negative log-probabilities as defined by Equation 6.

$$w(a) = -\log(p(a)) \quad (6)$$

In the tropical semiring, probability multiplication corresponds to weight addition and probability addition corresponds to weight maximization. In HFST-LEXC, we use OpenFST [1] as the software library for weighted finite-state transducers.

```

LEXICON Root
## CompoundNonFinalNoun ;
## CompoundFinalNoun ;

LEXICON Compound
0:# CompoundNonFinalNoun;
0:# CompoundFinalNoun;

LEXICON CompoundNonFinalNoun
isä  Compound "weight: -log(c(isä)/cs)" ;
isän Compound "weight: -log(c(isän)/cs)" ;
äiti Compound "weight: -log(c(äiti)/cs)" ;
äidin Compound "weight: -log(c(äidin)/cs)" ;

LEXICON CompoundFinalNoun
isä+sg+nom ## "weight: -log(c(isä+sg+nom)/cs)" ;
isä+sg+gen ## "weight: -log(c(isä+sg+gen)/cs)" ;
isä+sg+all ## "weight: -log(c(isä+sg+all)/cs)" ;
isä+pl+ins ## "weight: -log(c(isä+sg+all)/cs)" ;

LEXICON ##
## # ;

```

Figure 3: Structure weighting scheme using token penalties on the output language. Note that the functions in the comment field are placeholders for the actual weights.

For short, we call our unweighted compounding lexicon, *Lex*, and the decomposed noun compounding lexicon parts, i.e. the noun prefixes *CompoundNonFinalNoun\** in Figure 1 and the inflected noun forms *CompoundFinalNoun* in Figure 2, *Pref* and *Final*, respectively.

For an illustration of how the weighting scheme can be implemented in the weighted output language model, *WLex*, of the noun compounding lexicon, see Figure 3. There is an obvious extension of the weighting scheme to the output models of the decomposed unweighted lexicons, *Pref* and *Final*. We call these weighted output language models *WPref* and *WFinal*, respectively.

### 3.3 Back Off Model

The original lexicon, *Lex*, can be weighted by composing it with the weighted output language, *WLex*, as in Equation 7. However, there are a number of

word forms and compound segments in the lexicon, for which no estimate is available in the corpus. We wish to assign a large weight to these forms and segments, i.e. a weight  $M$  which is greater than any of the weights estimated from the corpus, e.g.  $M = \log(1 + cs)$ . To calculate the missing words, we first use the homomorphism  $uw$  to map the  $WPref$  to an unweighted automata, which we subtract from  $\Sigma^*$  and give the output model the final weight  $M$  using the homomorphism  $mw$ .

We create the following new sublexicons using automata difference and composition with the original decomposed transducers in Equations 8 and 9.

$$KnownAndSeenWords = Lex \circ WLex \quad (7)$$

$$MaxUnseenPref = Pref \circ (mw(\Sigma^* - uw(WPref))) \quad (8)$$

$$MaxUnseenFinal = Final \circ (mw(\Sigma^* - uw(WFinal))) \quad (9)$$

These sublexicons can be combined as specified in Equation 10 to cover the whole of the original lexicon.

$$\begin{aligned} WeightedLexicon = & KnownAndSeenWords \mid Pref \ MaxUnseenFinal \\ & \mid MaxUnseenPref \ Final \mid MaxUnseenPref \ MaxUnseenFinal \quad (10) \end{aligned}$$

The *WeightedLexicon* will assign the lowest corpus weight to the most likely reading and the highest corpus weight to the most unlikely reading of the original lexical transducer.

## 4 Data Sets

As training and test data, we use a compilation of three years, 1995-1997, of daily issues of Helsingin Sanomat, which is the most wide-spread Finnish newspaper. We disambiguated the corpus using Machineese for Finnish<sup>2</sup> which provided one reading in context for each word using syntactic parsing. This provided us with a mechanically derived standard and not a human controlled gold standard.

### 4.1 Training Data

The training data actually spanned 2.5 years with 1995 and 1996 of equal size and 1997 only half of this. This collection contained approximately 2.4 million different words, i.e. types, corresponding to approximately 70 million words of Finnish, i.e. tokens, divided into 29 million tokens for 1995, 29 for 1996 and 11 for 1997. We used the training data to count the non-compound tokens and their analyses.

### 4.2 Test Data

From the three years of training data we extracted running text from comparable sections of the news paper data. We chose articles from the section reporting on general news with normal running text (as a contrast to e.g. the

---

<sup>2</sup>Machineese is available from Connexor Ltd., [www.connexor.com](http://www.connexor.com)

economy or sports section with significant amounts of numbers and tables). The extracted test data sets contained 118 838, 134 837 and 193 733 tokens for 1995, 1996 and 1997, respectively. We used the test data to verify the result of the disambiguation.

### 4.3 Baseline

As a baseline method, we use the training data as such to create statistical unigram taggers as outlined in Section 3. In Table 1, we show the baseline result for the test data samples with a given training data tagger, the number of tokens with 1st correct reading, the number of tokens with some other correct reading, the number of tokens with some readings but no correct and the number of tokens with no reading.

Table 1: Baseline of the tagger test data.

Train Year	Test Year	1 <sup>st</sup>	$n^{th}$	No	No	Comment
		Correct (%)	Correct (%)	Correct (%)	Analysis (%)	
1995	1995	96.3	3.7	0.0	0.0	Max.
1995	1996	92.2	3.3	0.3	4.1	
1995	1997	91.9	3.3	0.3	4.6	
1996	1995	91.9	3.4	0.4	4.5	Max.
1996	1996	96.4	3.6	0.0	0.0	
1996	1997	92.4	3.2	0.3	4.1	
1997	1995	89.6	3.3	0.5	6.6	Max.
1997	1996	90.1	3.2	0.4	6.2	
1997	1997	96.7	3.3	0.0	0.0	

## 5 Tests and Results

We created two versions of the weighted lexicon for disambiguating running text. One weights the lexicon using the current corpus and tests the result using only the weighted lexicon data. The second test adds the baseline tagger to the lexicon in order to ensure some additional domain specific data for lack of a guesser.

### 5.1 Lexicon-based Unigram Tagger

We did our first tagging experiment using a full year of news paper articles as training data for the lexicon and testing with the test data from the other two years. The first correct results are consistently at 97 % of the words with some correct analysis. However, the coverage is totally dependent on the fairly restricted lexicon as shown in Table 2. We also include the results for testing and training on the same year as an upper limit or reference.

### 5.2 Extended Lexicon-based Unigram Tagger

We did our second tagging experiment as the first with the addition of using the full year of news paper data for extending the lexicon. Again, we tested with the test data from the other two years. The first correct results are consistently at 98 % of the words with some correct analysis and the coverage is now considerably better as shown in Table 3. We also include the results for testing and training on the same year as an upper limit or reference.

Table 2: Lexicon-based unigram tagger results for Finnish.

Train Year	Test Year	1 <sup>st</sup> Correct (%)	n <sup>th</sup> Correct (%)	No Correct (%)	No Analysis (%)	Comment
1995	1995	68.2	1.2	12.0	18.5	Max.
1995	1996	69.4	1.3	12.0	17.3	
1995	1997	69.4	1.4	11.7	17.5	
1996	1995	67.9	1.4	12.0	18.5	Max.
1996	1996	69.7	1.0	12.0	17.3	
1996	1997	69.4	1.3	11.7	17.5	
1997	1995	67.9	1.6	12.0	18.5	Max.
1997	1996	69.4	1.3	12.0	17.3	
1997	1997	69.6	1.3	11.7	17.5	

Table 3: Extended lexicon-based unigram tagger results for Finnish.

Train Year	Test Year	1 <sup>st</sup> Correct (%)	n <sup>th</sup> Correct (%)	No Correct (%)	No Analysis (%)	Comment
1995	1995	95.9	4.1	0.0	0.0	Max.
1995	1996	93.3	4.0	0.7	2.0	
1995	1997	93.1	4.0	0.6	2.3	
1996	1995	92.9	4.0	0.7	2.2	Max.
1996	1996	96.1	3.9	0.0	0.0	
1996	1997	93.6	3.7	0.6	1.9	
1997	1995	91.6	4.1	1.0	3.2	Max.
1997	1996	92.1	3.9	0.9	3.1	
1997	1997	96.3	3.7	0.0	0.0	

## 6 Discussion and Further Research

In this section we analyze the errors for which the correct tag sequence was not first, for which there was no correct tag sequence and for which there was no analysis at all. We present the most common by tag sequences or tokens. Finally, we make a few additional observations.

### 6.1 Correct Tag not 1<sup>st</sup> in Analysis

The cases where correct tag is not the first are dominated by the already known ambiguities where a token has multiple readings and both exist in corpus. One big class of these are verbs like *olla* or negation verb *ei*, since in perfect tense's passive the auxiliary is still in present tense active form (e.g. *on kerrottu* 'has been told' is *olla+pass+ind+pres kerrottu+pass+pcp2* while most likely reading of *on* 'is' is *olla+act+ind+pres+sg3*). The majority of variation between adjective readings and participles results also in number of wrong choices in tag strings with A or V PCP2. Also for many tokens the variation between adverb and adposition is purely syntactical and as such unigram tagger will fail in minority of cases. Also for handful of verbs, the A infinitive form falls together with 3<sup>rd</sup> person singular present tense (e.g. *järjestää* 'to arrange/(he) arranges') which causes fair amount of unigram tagger misreadings. The amount of compounds in analyses where first is not correct ranges from 6 % to 12 %.

Table 4: Error analysis for cases where correct reading exists.

Error Type	Baseline	Dictionary	Combined
Tagged 'ADV'/'PSP'	4112	2561	4331
Tagged 'A SG NOM'/'V PCP2 SG NOM'	3093	885	3388
Token 'on'	3855	0	3855
Token 'ei'	1170	0	1170
Token 'ollut'	735	0	735

## 6.2 Analyses without Correct Tag

For analyses where correct analysis was not among the readings, In corpus there’s a handful of underspecified analyses, such as (blah A), which aren’t produced at all by dictionary based analyzer, but assumably the corpus’s syntactic tagging mechanism has had use for those. Also for some adverbs and adpositions the dictionary only contains the non-lexicalised nominal form. There is also some overlap for cases in previous category here if the alternate reading for ambiguous form is not generated or found for some year.

Table 5: Error analysis for cases where correct reading is missing.

Error Type	Baseline	Dictionary	Combined
Tagged ‘A SG NOM’	24	771	82
Tagged ‘V PCP2 SG NOM’	20	4212	50
Tagged ‘A’	0	3300	0

## 6.3 No Analysis

For dictionary based tagger, the tokens which mainly dominate the missing analyses are proper nouns, abbreviations and numerals, which are known shortcomings for the analyzer. For other analyzers, such as baseline or extended, the main problem is proper nouns, many of which may appear only in one years issues. Also, since the dictionary based analyzer lacks productive numeral formation, many of the complex numeral expressions (e.g. *5–15-vuotiaat* ‘5-to-15-year-olds’) or specific numbers (e.g. *4029354*) are missing when using training corpora from one year to test other years analyses.

Table 6: Error analysis for cases where no results are given.

Error Type	Baseline	Dictionary	Combined
Proper nouns	17795	106101	17379
Token ‘klo’	0	13242	0
Token ‘mk’	0	5432	0
Tag NUM	699	7830	388

## 6.4 Other Observations

The error analysis confirms that the compounds for which the all parts were known contributed on the average 0.67 % to the overall error rate, i.e. correct not in the first position, for words with at least one correct analysis. For further discussions on the similarities and differences between Finnish, German and Swedish compounding, see [7].

If a disambiguated corpus is not available for calculating the word analysis probabilities, it is still possible to use only the string token probabilities to disambiguate the compound structure without saying anything about the most likely morphological reading. This segmentation would be similar to the segmentation the Morfessor software [3] tries to discover in an unsupervised way from corpora alone.

The good results for statistical morphological disambiguation of Finnish with a full morphological tag set using only a unigram model is most likely the result of the highly inflectional and compounding morphology of Finnish with free

word order. In order for a language to achieve a free word order, morphological ambiguities have to be resolvable locally almost without context.

As the inflected Finnish compounds correspond to noun phrases or prepositional phrases in English. This also sheds some additional light on the supposedly free word order in Finnish, which is similar to the rather free phrase ordering in many other languages, i.e. similar changes in the topic of a clause occurs in Finnish when shifting a phrase e.g. to a clause initial position.

## 7 Conclusions

We demonstrated how to build a weighted lexicon for a highly inflecting and compounding Fenno-Ugric language like Finnish. Similar methods apply to a number of Germanic languages with productive morphological compounding. From a practical point of view, we introduced the open source command line tools of HFST and used them successfully for compiling a weighted lexicon. We applied the weighted lexicon as a unigram tagger of running Finnish text achieving 97 % precision on words in the vocabulary. The unigram tagger is a good baseline when tagging morphologically complex languages like Finnish and for some purposes it may even be sufficient as such. In addition, it is easy to implement if a full-fledged morphological analyzer and a training corpus is available. For unknown foreign words and names, a guesser and an n-gram tagger may still be necessary.

## Acknowledgments

This research was funded by the Finnish Academy and the Finnish Ministry of Education. We are also grateful to the HFST–Helsinki Finite State Technology research team and to the anonymous reviewers for various improvements of the manuscript.

## References

- [1] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. <http://www.openfst.org>.
- [2] Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications. <http://www.fsmbook.com>.
- [3] Mathias Creutz, Krista Lagus, Krister Lindén, Sami Virpioja. 2005. Morphessor and Hutmegs: Unsupervised Morpheme Segmentation for Highly-Inflecting and Compounding Languages. In *Proceedings of the Second Baltic Conference on Human Language Technologies*.
- [4] Auli Hakulinen, Maria Vilkuna, Riitta Korhonen, Vesa Koivisto, Tarja Riitta Heinonen, and Irja Alho. 2008. *Iso suomen kielioppi*. Suoma-

laisen Kirjallisuuden Seura. referred on 31.12.2008, available from <http://scripta.kotus.fi/visk>.

- [5] Fred Karlsson. 1999. *Finns - An Essential Grammar*. Routledge. London. First published 1983 as *Finnish Grammar*.
- [6] Kimmo Koskeniemi. 1983. *Two-Level Morphology: A General Computational Model for Word Form Generation and Recognition*. *Publication No. 11. Publications of the Department of General Linguistics*. University of Helsinki.
- [7] Krister Lindén and Tommi Pirinen. 2009. *Weighted Finite-State Morphological Analysis of Finnish Compounding with HFST-LEXC*. In *Proceedings of NoDaLiDa 2009*.
- [8] Inari Listenmaa. 2009. *Combining Word Lists: Nykysuomen sanalista, Joukahainen-sanasto and Käänteissanakirja (in Finnish)*. Bachelor's Thesis. Department of Linguistics. University of Helsinki.
- [9] Torsten Marek. 2006. *Analysis of German Compounds using Weighted Finite State Transducers*. Technical report, Eberhard-Karls-Universität Tübingen.
- [10] Tommi Pirinen. 2008. *Suomen kielen äärellistilainen automaattinen morfologinen analyysi avoimen lähdekoodin keinoin*. Master's thesis, Helsingin yliopisto.
- [11] Anne Schiller. 2005. *German Compound Analysis with wfsc*. In *FSMNL*, pages 239–246.