

Using a Morphological Database to Increase the Accuracy in POS Tagging

Hrafn Loftsson

School of Computer Science
Reykjavik University
hrafn@ru.is

Sigrún Helgadóttir

The Árni Magnússon Institute
for Icelandic Studies
sigruhel@hi.is

Eiríkur Rögnvaldsson

Department of Icelandic
University of Iceland
eirikur@hi.is

Abstract

We experiment with extending the dictionaries used by three open-source part-of-speech taggers, by using data from a large Icelandic morphological database. We show that the accuracy of the taggers can be improved significantly by using the database. The reason is that the unknown word ratio reduces dramatically when adding data from the database to the taggers' dictionaries. For the best performing tagger, the overall tagging accuracy increases from the base tagging result of 92.73% to 93.32%, when the unknown word ratio decreases from 6.8% to 1.1%. When we add reliable frequency information to the tag profiles for some of the words originating from the database, we are able to increase the accuracy further to 93.48% – this is equivalent to 10.3% error reduction compared to the base tagger.

1 Introduction

In general, part-of-speech (PoS) taggers can be categorised into two types. First, *data-driven* taggers, i.e. taggers that are trained on pre-tagged corpora and are both language and tagset independent, e.g. (Brants, 2000; Toutanova et al., 2003; Shen et al., 2007). Second, *linguistic rule-based taggers*, which are developed “by hand” using linguistic knowledge, with the purpose of tagging a specific language using a particular tagset, e.g. (Karlsson et al., 1995; Loftsson, 2008).

All taggers use a particular tagset T and rely on a dictionary D containing the *tag profile* (ambiguity class) T_w for each word w . A tag profile T_w indicates which tags are assignable to w , thus $T_w \subset T$. Essentially, for each word w , a tagger disambiguates T_w by selecting (or removing all but) one tag from it with regard to context. The

dictionary D is derived by a data-driven tagger during training, and derived or built during development of a linguistic rule-based tagger.

When tagging new text, PoS taggers frequently encounter words that are not in D , i.e. so-called *unknown words*. An unknown word u can be quite problematic for a tagger, because the tag profile for u needs to be guessed. In most cases, PoS taggers therefore contain a special module, called an *unknown word guesser*, to generate the tag profile for unknown words. Frequently, the guessing of the tag profile for unknown words is incorrect and therefore the tagging accuracy for these words is considerably lower than the tagging accuracy for known words. To increase the overall tagging accuracy of PoS taggers, one might therefore try to refine the underlying unknown word guessers. Another approach is simply to try to minimise the ratio of unknown words by extending the dictionaries used by the taggers.

In this paper, we use the latter approach. We experiment with extending the dictionaries used by three PoS taggers for Icelandic with data from a large morphological database (Bjarnadóttir, 2005). Our logical assumption is that the overall tagging accuracies of the taggers can be increased by this method, but we are also interested in how extended dictionaries affect the accuracy for unknown words and known words separately.

The three taggers used in our experiments are: i) the linguistic rule-based tagger *IceTagger* (Loftsson, 2008); ii) *TriTagger*, a re-implementation of the statistical tagger *TnT* by Brants (2000); and iii) a serial combination of the two (Loftsson et al., 2009).

The morphological database does not contain any frequency information for the tags in the tag profile for each word, but, nevertheless, we show that the tagging accuracy of the taggers can be improved significantly by using the database. The reason is that when we add most of the data from

the database to the taggers’ dictionaries the unknown word ratio decreases dramatically, from 6.8% to 1.1%. In that case, the overall tagging accuracy of the best performing tagger, the serial combination of IceTagger and TriTagger, increases from the base tagging result of 92.73% to 93.32%. When we add reliable frequency information, derived from a corpus, to the tag profiles for a part of the words originating from the database, we are able to increase the accuracy further to 93.48% – this is equivalent to 10.3% error reduction compared to the base tagger.

Interestingly, it seems that very few papers exist in the literature regarding extensions of the dictionaries used by PoS taggers. In (Rupnik et al., 2008), a dictionary derived from training is essentially extended by using a backup lexicon extracted from a large corpus (which is different from the training corpus). In contrast, we use a morphological database to extend a tagger’s dictionary, but use a corpus for deriving frequency information for part of the dictionary entries. In (Tufis et al., 2008), an unknown word u , and its tag profile and lemma obtained by a tagger when tagging new texts, is used by a morphological generator to generate tag profiles for new word forms that are morphologically related to u . The dictionary is thus extended incrementally, each time new text is tagged. In contrast, since we have access to a large morphological database, we extend a tagger’s dictionary once and for all.

2 The morphological database

At the Árni Magnússon Institute for Icelandic Studies, a comprehensive full form database of modern Icelandic inflections has been developed (Bjarnadóttir, 2005). Its Icelandic abbreviation is *BÍN* (“Beygingarlýsing íslensks nútímamáls”), and henceforth we use that term. *BÍN* contains about 280,000 paradigms, with over 5.8 million inflectional forms. The output from the database used in this project contains lemma, word form, word class, and morphological features for common nouns, proper nouns, adjectives, verbs, and adverbs. It is important to note that the database does, however, not contain any frequency information for the word forms.

A web interface for *BÍN* is available at <http://bin.arnastofnun.is>, from where a text file in the format used in this project can be downloaded. Below are 16 lines from the file, demon-

strating entries for the lemma “hestur” ‘horse’:

```
hestur;6179;kk;alm;hestur;NFET
hestur;6179;kk;alm;hesturinn;NFETgr
hestur;6179;kk;alm;hest;PFET
hestur;6179;kk;alm;hestinn;PFETgr
hestur;6179;kk;alm;hesti;PGFET
hestur;6179;kk;alm;hestinum;PGFETgr
hestur;6179;kk;alm;hests;EFET
hestur;6179;kk;alm;hestsins;EFETgr
hestur;6179;kk;alm;hestar;NFFT
hestur;6179;kk;alm;hestarnir;NFFTgr
hestur;6179;kk;alm;hesta;PFFT
hestur;6179;kk;alm;hestana;PFFTgr
hestur;6179;kk;alm;hestum;PGFFT
hestur;6179;kk;alm;hestunum;PGFFTgr
hestur;6179;kk;alm;hesta;EFFT
hestur;6179;kk;alm;hestanna;EFFTgr
```

The exact meaning of the data in each column is not important for our discussion, but we point out that the lemma is in the first column, gender is in third column (“kk”=masculine), the word form is in the fifth column, and the morphological features case, number and definiteness are in the last column (for example, “NF”=nominative, “ET”=singular, “gr”=definite article).

3 The corpus and the taggers used

The Icelandic Frequency Dictionary (IFD) corpus (Pind et al., 1991) has been used to train and test taggers for Icelandic (Helgadóttir, 2005; Loftsson, 2008; Dredze and Wallenberg, 2008; Loftsson et al., 2009). The corpus contains about 590,000 tokens, and its underlying tagset about 700 tags, of which 639 tags actually appear in the corpus. The tags are character strings where each character has a particular function. The first character denotes the *word class*. For each word class there is a predefined number of additional characters (at most six), which describe morphological features, like *gender*, *number* and *case* for nouns; *degree* and *declension* for adjectives; *voice*, *mood* and *tense* for verbs, etc. To illustrate, consider the word form “hestur” ‘horse’. The corresponding tag is “nken”, denoting noun (n), masculine (k), singular (e), and nominative (n) case.

As mentioned in Section 1, we use one linguistic rule-based tagger (IceTagger), one data-driven tagger (TriTagger), and a serial combination of the two in our experiments. Both IceTagger and TriTagger are implemented in Java and are part of the open-source IceNLP toolkit¹.

IceTagger is reductionistic in nature, i.e. it removes inappropriate tags from the tag profile T_w

¹IceNLP is available at <http://icenlp.sourceforge.net>

for a specific word w in a given context. IceTagger first applies local rules for initial disambiguation and then uses a set of heuristics (global rules) for further disambiguation. The tag profile for each word used by IceTagger is ordered by the frequency of the tags – the first tag listed is the most frequent one and the last tag is the least frequent one. If a word is still ambiguous after the application of the heuristics, the default heuristic is simply to choose the most frequent tag (the first tag) for the word. An important part of IceTagger is its unknown word guesser, *IceMorph*. It guesses the tag profile for unknown words by applying morphological analysis and ending analysis. In addition, *IceMorph* can fill in the *tag profile gaps*² in the dictionary for words belonging to certain morphological classes (Loftsson, 2008).

TriTagger is a re-implementation of the well known Hidden Markov Model (HMM) tagger TnT by Brants (2000)³. TriTagger uses a trigram model to find the sequence of tags for words in a sentence which maximises the product of contextual probabilities ($P(t_i|t_{i-2}, t_{i-1})$) and lexical probabilities ($P(w_i|t_i)$):

$$P(t_1)P(t_2|t_1) \prod_{i=3}^n P(t_i|t_{i-2}, t_{i-1}) \prod_{i=1}^n P(w_i|t_i) \quad (1)$$

In the above equation, w_i denotes word i in a sentence of length n ($1 \leq i \leq n$) and t_i denotes the tag for w_i . The probabilities are derived using maximum likelihood estimation based on the frequencies of tags found during training.

HMM taggers handle unknown words by setting tag probabilities according to words’ suffixes. The term suffix is here defined as a final sequence of characters of a word. TnT, and thus TriTagger, generate probability distributions for suffixes of various lengths. The distribution for particular suffixes is based on words in the training data that share the same suffix. The reader is referred to (Brants, 2000) for the details of suffix handling.

²A tag profile gap for a word occurs when a tag is missing from the tag profile. This occurs, for example, if not all possible tags for a given word are encountered during training.

³The TnT tagger is extremely efficient – both training and testing are very fast. Unfortunately, TnT is closed source which limits its use when changes need to be carried out to its default behaviour. TriTagger is open-source and therefore its functionality can be changed or extended relatively easily. Moreover, our experiments have shown that its tagging accuracy is almost identical to the accuracy obtained by TnT. On the other hand, TriTagger has not been optimised for run-time efficiency.

Below, we exemplify the tag profiles stored in the dictionaries for IceTagger and TriTagger for a specific word “konu” ‘woman’:

```
konu nveþ nveo nvee
konu 122 nveþ 44 nveo 42 nvee 36
```

The first tag profile is stored in the dictionary for IceTagger. The possible tags are “nveþ”, “nveo”, and “nvee” (denoting noun, feminine, singular, dative/accusative/genitive), sorted by decreasing frequency. The second tag profile is stored in the dictionary for TriTagger. It contains similar information, but, additionally, frequency information is attached to both the word itself and each possible tag.

3.1 Base tagging results

We have previously shown (Loftsson et al., 2009) that a significant improvement in tagging accuracy is obtainable by running a serial combination of IceTagger and a HMM tagger (TriTagger). Specifically, the best result was obtained by making the HMM perform initial disambiguation only with regard to the word class (the first letter of a tag), then running IceTagger, and finally by making the HMM disambiguate words that IceTagger was not able to fully disambiguate. This tagger is called *HMM+Ice+HMM*.

In our current experiments, we use 10-fold cross-validation on the exact same training and test splits of the so-called *corrected version* of the IFD corpus used by Loftsson et al. (2009). Each test corpus contains about 10% of the tokens from the IFD, while the corresponding training corpus contains about 90% of the tokens. The average unknown word ratio using this data split is about 6.8%.

We use a version of the corrected IFD corpus in which type information for proper nouns (named-entity classification) has been removed, and additionally we only use one tag for numerical constants. The reason for these changes is to make the tagset of the corpus comparable to tagsets for other languages. These changes reduce the size of the tagset from about 700 tags to about 600 tags, and the number of tags actually appearing in the IFD reduces from 639 tags to 567.

Table 1 shows the average accuracy of the three taggers. In this table (and in all the ones that follow), the average accuracy is based on testing using the first nine test corpora, because the tenth one was used for developing IceTagger. We consider the accuracy figures in Table 1 as our base

Tagger	Unknown	Known	All
TriTagger	72.98	92.18	90.86
IceTagger	77.02	93.07	91.98
HMM+Ice+HMM	77.47	93.84	92.73

Table 1: Average base tagging accuracy (%). Average ratio of unknown words in testing is 6.8%.

tagging results – in the experiments described in the next section we try to improve on these figures.

4 The experiments

In this section, we describe the setup and results of two experiments. First, we extend the dictionaries used by the three taggers by using data from the morphological database BÍN. Second, we add reliable frequency information to some of the dictionary entries (tag profiles).

4.1 Extending the dictionaries

This part of our experiment is in two parts. First, we generate a file F_1 by extracting only lemmata from the database output described in Section 2. F_1 contains about 280,000 lemmata. To clarify, only the first line in the example output shown in Section 2 is then included in F_1 . Second, we drop the lemmata condition and generate a file F_2 by selecting most of the word forms from the database output⁴. F_2 contains about 5.3 million rows.

To generate an extended dictionary for a tagger (classifier) C using data from F_1 , we perform the following (the same procedure applies when using F_2):

1. Derive a dictionary from F_1 , containing words and their corresponding tag profiles. Symbols denoting morphological features in F_1 are mapped to the symbols used in the IFD tagset. We call the resulting dictionary D_{BIN} .
2. Combine D_{BIN} with the dictionary D generated by a tagger C during training (the number of entries in D are about 55,000, on the average). The result is a new dictionary D_{EXT} . If a word exists in both D and D_{BIN} then only the entry from D appears in D_{EXT} .
3. Test tagger C using dictionary D_{EXT} .

⁴Because of memory issues with the taggers, we exclude proper nouns that are names of places.

Tagger	Unknown	Known	All
TriTagger	74.44	91.53	90.63
IceTagger	80.44	92.83	92.18
HMM+Ice+HMM	80.53	93.57	92.89

Table 2: Average tagging accuracy (%) using dictionaries extended with lemmata only from BÍN. Average ratio of unknown words in testing is about 5.3%.

The above description holds when generating an extended dictionary for IceTagger, a tagger which does not need frequency information in the tag profile for words. In the case of TriTagger, we simply assume a uniform distribution, i.e. we mark each tag in the tag profile T_w for word w with the frequency 1. Note that for TriTagger, extending the dictionary only affects the lexical probabilities from Equation 1 – the contextual probabilities remain unchanged.

Recall (from Section 3) that HMM taggers handle unknown words by generating probability distributions for suffixes of various lengths using the words in the training data. We want the generation of these probability distributions to be only dependent on the data from D (from the IFD corpus), but not as well from D_{BIN} . The reason is twofold. First, the IFD corpus is large enough for deriving reliable suffix probability distributions. Second, using all the words from a very large dictionary (like D_{EXT}) to generate the distributions significantly slows down the tagging process. This issue demonstrates the importance of having access to open-source software. We simply changed the loading module of TriTagger such that it does not use all dictionary entries for suffix handling. If the loading module finds a special entry in the dictionary (essentially a specially marked comment) it does not use the succeeding entries for suffix handling. We put the special entry into D_{EXT} after the last entry from D and thus before the first entry from D_{BIN} .

Let us first consider the case of using file F_1 for extending the dictionaries, i.e. when only extracting lemmata from the database output. In that case, the resulting D_{BIN} contains about 260,000 entries. Table 2 shows the accuracy of the taggers when using this version of the extended dictionary.

Comparing the results from Tables 2 and 1, we note the following:

- The average unknown word ratio decreases

by about 1.5% (from about 6.8% to about 5.3%).

- The accuracy for known words decreases in the three taggers. The most probable reason is that the tag profile for some of the lemmata entries coming from D_{BIN} contains gaps (see Section 3). This can be attributed to the fact that only a single line from the database output is selected when extracting the lemmata, but in many cases a lemma can have multiple analysis (tags). Note that this decrease in accuracy for known words is considerably higher in TriTagger (0.65 percentage points) than in IceTagger (0.24 percentage points). This is because the unknown word guesser IceMorphy, used by IceTagger, can fill into the tag profile gaps for certain morphological classes, as mentioned in Section 3.
- The accuracy for unknown words increases in all the three taggers – the highest gain (3.42 percentage points) is obtained by IceTagger. For the case of IceTagger the reason is that IceMorphy first applies morphological analysis to unknown words (before trying ending analysis). For an unknown word u , IceMorphy searches for a morphologically related word (a known word) to u in its dictionary, i.e. a word containing the same stem but a different morphological suffix. The added lemmata entries can thus serve as related words for unknown words and since the morphological analysis module of IceTagger is quite accurate (Loftsson, 2008), the added lemmata entries help to increase the tagging accuracy of unknown words.
- The accuracy for all words increases in both IceTagger and HMM+Ice+HMM, but only by 0.20 and 0.16 percentage points, respectively. Obviously, the decreased accuracy for known words “cut backs” the gain obtained in the accuracy for unknown words. TriTagger’s relatively large reduction in accuracy for known words is to blame for the reduction in its accuracy for all words.

Let us now consider the second case, when using file F_2 for extending the dictionaries. F_2 contains most of the entries from the database and the resulting D_{BIN} contains about 2.6 million entries.

Tagger	Unknown	Known	All
TriTagger	65.82	91.96	91.66
IceTagger	63.38	92.86	92.53
HMM+Ice+HMM	60.41	93.69	93.32

Table 3: Average tagging accuracy (%) using dictionaries extended with most of the data from BÍN. Average ratio of unknown words in testing is 1.1%.

Table 3 shows the accuracy of the taggers when using this large version of the extended dictionary.

Comparing the results from Tables 3 and 1, we note the following:

- The average unknown word ratio drops down to 1.1%. Concurrently, the accuracy for unknown words decreases substantially in all the three taggers. This is because the unknown word ratio drops dramatically and only “hard” unknown words remain – mostly proper nouns and foreign words.
- The accuracy for known words decreases in the three taggers by 0.15-0.22 percentage points. This is a lower decrease than when using only lemmata entries from BÍN (see Table 2) and can be explained by the fact that in this case the added entries from BÍN should not contain tag profile gaps. Why do we then see a slight decrease in accuracy for known words? Recall that BÍN does not contain any frequency information and therefore, for the added dictionary entries, we had to: i) assume a uniform distribution of tags in the the tag profile for TriTagger, and ii) assume no specific order for the tags in the tag profile for IceTagger (see the discussion on the order of the tags in Section 3). This is the most probable reason for the slight reduction in the tagging accuracy of known words.
- The accuracy for all words increases significantly in all the three taggers, about 0.4-0.8 percentage points. This result confirms our logical assumption that the tagging accuracy can be increased by extending the dictionaries of taggers – even in the absence of reliable frequency information.

4.2 Adding frequency information

Recall from Section 3 that the tag profile in the dictionary used by IceTagger is assumed to be

sorted. When a word cannot be fully disambiguated, this enables IceTagger to select the most frequent tag (the first tag) in the tag profile for the word. On the other hand, when frequency information is missing, as is the case for the BÍN data, the first tag of the remaining tags in the tag profile may or may not be the most frequent tag. Thus, when IceTagger applies the default heuristic to choose the first tag that may be an arbitrary choice.

For a HMM tagger, the lack of reliable frequency information in a tag profile for a word can also cause problems. This follows directly from Equation 1, i.e. the term $P(w_i|t_i)$ stands for lexical probabilities which are computed using maximum likelihood estimation from a dictionary containing frequency information for each tag in the tag profiles for words.

In order to get reliable frequency information for the BÍN data, we use a tagged corpus named MÍM (“Mörkuð íslensk málheild”; <http://mim.hi.is>) which is being developed at the Árni Magnússon Institute for Icelandic Studies. The final size of the MÍM corpus will be 25 million tokens, but the version that we use contains about 17 million tokens.

Recall from Section 4.1 that D_{BIN} denotes a dictionary derived from BÍN. From the MÍM corpus, we derive a frequency dictionary D_{MIM} . We then create a new dictionary D_{NEW} (based on D_{BIN}) in which frequency information for some of its tag profiles comes from D_{MIM} . Specifically, we use the following procedure:

1. Each word w in D_{BIN} is looked up in D_{MIM} . If w is not found in D_{MIM} , then w and its tag profile is copied to D_{NEW} . Each tag in the tag profile for w is given the frequency 1 (i.e. a uniform distribution is assumed). If w is found in D_{MIM} , proceed to step 2.
2. Order the tags in the tag profile for w in D_{BIN} , according to the frequencies of the tags in the tag profile for w in D_{MIM} . If a tag t for a word w is found in D_{MIM} but not in D_{BIN} , then t does not become a part of the tag profile for w in D_{NEW} . The reason is that the dictionary D_{MIM} is derived from a tagged corpus which has not been manually inspected and thus contains tagging errors. In other words, the tag profile from D_{BIN}

Tagger	Unknown	Known	All
TriTagger	65.84	92.22	91.93
IceTagger	63.47	93.11	92.78
HMM+Ice+HMM	60.50	93.85	93.48

Table 4: Average tagging accuracy (%) using dictionaries extended with most of the data from BÍN and with arranged tag profiles for some of the words. Average ratio of unknown words in testing is 1.1%.

is considered more reliable than the one in D_{MIM} .

3. Combine the new dictionary D_{NEW} with the dictionary D used by a tagger C as explained in step 2 in Section 4.1.

To illustrate, consider the following three tag profiles for the word “skögultennur” ‘buckteeth’:

```
skögultennur nvfn nvfo
skögultennur nvfo nken nvfn
skögultennur nvfo nvfn
```

The first tag profile appears in D_{BIN} . The tags “nvfn” and “nvfo” appear in alphabetic order. The second tag profile appears in D_{MIM} (shown here without the frequency numbers for each tag). The tag profile is sorted in ascending order of frequency of the tags. Note that the second tag profile contains the tag “nken” (resulting from a tagging error in MÍM) which does not appear in the first tag profile. When generating the resulting tag profile for D_{NEW} – the third line in the illustration above – the tag “nken” does thus not appear.

We used the procedure described above to generate extended dictionaries with frequency information for TriTagger and sorted tag profiles for IceTagger. Of the 2.6 million tag profiles in D_{BIN} , 250,000 were found in D_{MIM} (i.e. about 10%). This procedure thus “arranged” 250,000 of the tag profiles in D_{BIN} .

Table 4 shows the result of using the three taggers with extended dictionaries and with arranged tag profiles for some of the words. The accuracy of TriTagger improves from 91.66%, when using BÍN data without frequency information (see Table 3) to 91.93% (3.25% error reduction). The accuracy of IceTagger improves from 92.53% to 92.78% (3.5% error reduction), and the accuracy of HMM+Ice+HMM improves from 93.32% to 93.48% (2.4% error reduction). The error reduction between our HMM+Ice+HMM tagger, with

an extended dictionary and arranged tag profiles, and the base version of HMM+Ice+HMM (see Table 1), is 10.3%.

5 Future work

In Section 4.2, we showed that the accuracies of the three taggers can be improved significantly by arranging the tag profiles of the taggers using frequency information from the MÍM corpus. We used about 17 million tokens from the corpus, but once it has been extended to its final size of 25 million tokens, we would like to repeat this part of the experiment, thus using more data, to see if the accuracy increases further.

Note that we have only been able to arrange part of the tag profiles (about 10%) in the extended dictionaries by using frequency information from MÍM. In future work, we would also like to experiment with arranging the remainder of the tag profiles according to unigram tag frequencies (for example, derived from the IFD corpus), i.e. tag frequencies that are not associated with individual words. We would then be seeking an answer to the question whether assigning unigram tag frequencies to the tag profiles of words, for which we do not have reliable frequency information, results in higher tagging accuracy compared to assigning a uniform distribution to the tag profiles (i.e. giving each tag the frequency 1 as we have done).

6 Conclusion

We have experimented with adding data from a large morphological database to the dictionaries used by three open-source PoS taggers for Icelandic. Our results show that the tagging accuracy improves significantly when extending the dictionaries, and even further improvement in accuracy can be obtained by adding frequency information to some of the dictionary entries (tag profiles).

Our best performing tagger, a serial combination of a linguistic rule-based tagger and a statistical tagger, obtains a state-of-the-art tagging accuracy of 93.48% when using extended dictionaries and added frequency information. This is equivalent to 10.3% error reduction compared to the best base tagger.

Acknowledgments

The work presented in this paper was partly supported by the Icelandic Research Fund, grant 070025023.

References

- K. Bjarnadóttir. 2005. Modern Icelandic Inflections. In H. Holmboe, editor, *Nordisk Sprogteknologi 2005*. Museum Tusulanums Forlag, Copenhagen.
- T. Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, Seattle, WA, USA.
- M. Dredze and J. Wallenberg. 2008. Icelandic Data Driven Part of Speech Tagging. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, OH, USA.
- S. Helgadóttir. 2005. Testing Data-Driven Learning Algorithms for PoS Tagging of Icelandic. In H. Holmboe, editor, *Nordisk Sprogteknologi 2004*. Museum Tusulanums Forlag, Copenhagen.
- F. Karlsson, A. Voutilainen, J. Heikkilä, and A. Anttila. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin, Germany.
- H. Loftsson, I. Kramarczyk, S. Helgadóttir, and E. Rögnvaldsson. 2009. Improving the PoS tagging accuracy of Icelandic text. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA-2009)*, Odense, Denmark.
- H. Loftsson. 2008. Tagging Icelandic text: A linguistic rule-based approach. *Nordic Journal of Linguistics*, 31(1):47–72.
- J. Pind, F. Magnússon, and S. Briem. 1991. *Íslensk orðtúðnibók [The Icelandic Frequency Dictionary]*. The Institute of Lexicography, University of Iceland, Reykjavik, Iceland.
- J. Rupnik, M. Grčar, and T. Erjavec. 2008. Improving Morphosyntactic Tagging of Slovene Language through Meta-tagging. *Informatica*, 32(4):437–444.
- L. Shen, G. Satta, and A. Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.
- K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT/NAACL 2003*, Edmonton, Canada.
- D. Tufis, E. Irimia, R. Ion, and A. Ceausu. 2008. Un-supervised Lexical Acquisition for Part of Speech Tagging. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.