Lang Resources & Evaluation
DOI 10.1007/s10579-015-9326-3

PROJECT NOTES

FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish

Miikka Silfverberg[1] · Teemu Ruokolainen[2] · Krister Lindén[1] · Mikko Kurimo[2]

Abstract This paper describes FinnPos, an open-source morphological tagging and lemmatization toolkit for Finnish. The morphological tagging model is based on the averaged structured perceptron classifier. Given training data, new taggers are estimated in a computationally efficient manner using a combination of beam search and model cascade. The lemmatization is performed employing a combination of a rule-based morphological analyzer, OMorFi, and a data-driven lemmatization model. The toolkit is readily applicable for tagging and lemmatization of running text with models learned from the recently published Finnish Turku Dependency Treebank and FinnTreeBank. Empirical evaluation on these corpora shows that FinnPos performs favorably compared to reference systems in terms of tagging and lemmatization accuracy. In addition, we demonstrate that our system is highly competitive with regard to computational efficiency of learning new models and assigning analyses to novel sentences.

Keywords Morphological tagging · Data-driven lemmatization · Averaged perceptron · Finnish · Open-source

1 Introduction

This paper presents FinnPos, an open-source morphological tagging and lemmatization toolkit for Finnish. Our work stems from the recently published Turku Dependency Treebank (Haverinen et al. 2009, 2014) and FinnTreeBank

✉ Miikka Silfverberg
mpsilfve@iki.fi

[1] University of Helsinki, Helsinki, Finland

[2] Aalto University, Helsinki, Finland

Published online: 14 December 2015

(Voutilainen 2011). The Turku Dependency Treebank has been prepared by manually correcting the output of an automatic annotation process, whereas the FinnTreeBank has been prepared completely manually. These data sets are the first treebanks published for Finnish and are freely available for research purposes.

The morphological tagging component of the FinnPos system is based on the well-known averaged structured perceptron classifier (Collins 2002). We accelerate perceptron learning using a combination of two approximations, namely, beam search and a model cascade inspired by the work of Müller et al. (2013) on the conditional random field (CRF) model. Approximative learning is necessary since exact model estimation is infeasible given the vast number of morphological labels present in the data. Meanwhile, the lemmatization is performed using OMorFi, an open-source morphological analyzer for Finnish (Pirinen 2008). In order to lemmatize word forms unknown to OMorFi, FinnPos implements a statistical lemmatization model which learns to perform lemmatization for any given word form in a data-driven manner following Chrupala et al. (2008).

Earlier work on morphological tagging of Finnish has utilized rule-based methodology, exemplified by the Constraint Grammar approach of Karlsson (1990). Due to the recentness of the corpora applicable for learning, there exists relatively little work on statistical morphological tagging of Finnish. An exception is the work of Silfverberg and Linden (2011) who investigated a finite state machine implementation of the classic hidden Markov model tagging approach of Brants (2000). More recently, Bohnet et al. (2013) investigated joint morphological tagging and dependency parsing of Finnish on the Turku Dependency Treebank. In their work, the morphological tagging was performed using the MarMot system (Müller et al. 2013).

The main contributions of this work are as follows. First, we present the first morphological tagging and lemmatization toolkit designed specifically for Finnish. The presented toolkit, FinnPos, is provided as an open-source implementation[1]. Second, we compare the FinnPos system to three established morphological analysis toolkits, namely, HunPos (Halácsy et al. 2007), Morfette (Chrupala et al. 2008), and MarMot (Müller et al. 2013). The performed empirical evaluation shows that FinnPos performs favorably to the reference systems in terms of tagging and lemmatization accuracy, as well as computational efficiency of learning new models and assigning analyses to novel sentences.

The rest of the paper is organized as follows. In Sect. 2, we describe the treebanks employed for training and evaluation of the FinnPos system. The methods implemented by the FinnPos system are discussed in Sect. 3. In Sect. 4, we present the empirical evaluation. Finally, conclusions on the work are presented in Sect. 5.

2 Treebanks

This section describes the treebanks employed for training and evaluation of the FinnPos system.

[1] https://github.com/mpsilfve/FinnPos

Table 1 A disambiguated analysis for an exemplar sentence *Hän ei asu pienessä kylässä* using Turku Treebank annotation adapted from Haverinen et al. (2014)

Word form	Lemma	POS	Other tags
Hän	hän	Pronoun	pers sg nom up
ei	ei	Verb	neg sg3 act
asu	asua	Verb	prs ind conneg
pienessä	pieni	Adjective	sg ine pos
kylässä	kylä	Adverb	sg ine

Table 2 Summary of Turku Dependency Treebank (TDT) (Haverinen et al. 2009, 2014) and Finn-TreeBank (FTB) (Voutilainen 2011)

	TDT	FTB
Size	13,572 sent. (183,118 tok.)	19,121 sent. (162,028 tok.)
# labels	2014	1399
OMorFi coverage	94.2 %	99.0 %

The OMorFi coverage refers to coverage per token

Turku Dependency Treebank.

The Turku Dependency Treebank (TDT) (Haverinen et al. 2009, 2014) contains text from ten varying domains, such as Wikipedia articles, blog entries, and financial news. The annotation has been prepared by manually correcting the output of an automatic annotation process. The morphological analyses of word tokens are post-processed outputs of OMorFi, an open-source morphological analyzer for Finnish (Pirinen 2008). The resulting TDT annotation for each word token consists of word lemma (base form), part-of-speech (POS), and a list of detailed morphological information, including case, number, tense, and so forth. Table 1 shows the analysis of an exemplar sentence *Hän ei asu pienessä kylässä* (*(S)he doesn't live in a small village*). We refer to the combination of the POS and the more specific tags as the *morphological label*.

FinnTreeBank.

The FinnTreeBank (FTB) (Voutilainen 2011) is a morphologically tagged and dependency parsed collection of example sentences from Iso Suomen Kielioppi, a descriptive grammar of the Finnish language (Hakulinen et al. 2004). Similarly to TDT, FTB contains text from various domains including newspapers and fiction. The major difference between TDT and FTB, therefore, is that FTB contains a variety of grammatical examples, whereas TDT contains more real-life language use. Both the morphological tagging and dependency structures have been manually prepared.

Similarly to TDT, the morphological analyses of word tokens in FTB are post-processed outputs of OMorFi (Pirinen 2008). However, the treebanks are based on different versions of OMorFi. Moreover, the post-processing steps applied in TDT and FTB differ. This results in somewhat different annotation schemes. Finally, for a summarizing presentation of TDT and FTB, see Table 2.

3 Methods

In the FinnPos system, we regard the morphological tagging and lemmatization tasks as two separate sub-problems. Given a sentence, each word form is assigned a morphological label by the morphological tagger based on the averaged structured perceptron (Collins 2002). Subsequent to assigning the morphological label, selecting the appropriate word lemma is, in principle, straightforward given the set of full analyses (morphological labels and lemmas) provided by the OMorFi analyzer. However, OMorFi does not have full vocabulary coverage, that is, for some word forms no analyses are returned. In these cases, a simple baseline solution would be to simply return the original word form as the lemma. However, a more appealing approach is to learn a lemmatization model in a data-driven manner and apply it to lemmatize the unknown word forms (Chrupala et al. 2008). In what follows, we describe the applied structured perceptron tagger and data-driven lemmatizer in Sects. 3.1 and 3.2, respectively.

3.1 Averaged structured perceptron

In this section, we describe the morphological tagging component based on the averaged structured perceptron classifier presented originally for sequence labeling by Collins (2002). The issues covered include the model definition, feature extraction, model estimation from training data, and decoding.

3.1.1 Scoring function

Given a sentence $x = (x_1, \ldots, x_{|x|})$ and a label sequence $y = (y_1, \ldots, y_{|x|})$, the structured perceptron classifier assigns the pair (x, y) a score

$$\mathrm{score}(\mathrm{x}, \mathrm{y}; \boldsymbol{w}) = \sum_{i=n}^{|x|} \boldsymbol{w} \cdot \boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i) \,, \tag{1}$$

where n denotes the model order, $\boldsymbol{w}$ the model parameter vector, and $\boldsymbol{\phi}$ the feature extraction function. The word forms x_i are assigned labels from a potentially large label set $\mathcal{Y}$, that is, $y_i \in \mathcal{Y}$ for all $i = 1, \ldots, |x|$. As shown in Table 2, for TDT and FTB, the label sets $\mathcal{Y}$ contain roughly 2000 and 1400 morphological tags, respectively.

3.1.2 Feature extraction

The appeal of the perceptron classifier (1) lies in its capability of utilizing rich, overlapping feature sets. The individual features correspond to the elements of feature vector $\boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i)$. In the FinnPos system, the feature extraction scheme follows the *node-observation* presentation of Sutton and McCallum (2011), in which each label position is associated with a set of features describing the input. Specifically, we follow the classic work of Ratnaparkhi (1996) on morphological tagging and include the following input feature set:

1. Bias (always active irrespective of input).
2. Word forms $x_{i-2}, \ldots, x_{i+2}$.
3. Prefixes and suffixes of the word form x_i up to length δ_{affix}.
4. If the word form x_i contains (one or more) capital letter, hyphen, dash, or digit.

In addition, we use the following binary functions:

5. The lower-cased word form x_i.
6. The word pairs (x_{i-1}, x_i) and (x_i, x_{i+1}).

When using a morphological analyzer, we also include:

7. Each morphological label of word x_i.

In addition, the node-observation scheme utilizes label transition features to capture the fact that some label transitions occur more often than others. For example, the Finnish word *asu* could occur with a noun (*clothing*) or verb (a negative or imperative form of *to live*) label. In the example in Table 1, it is, however, preceded by a negator (*ei*). Therefore, in this context, *asu* is more likely to be a verb rather than a noun since in Finnish negators seldomly precede nouns.

The above features treat the morphological labels as single entities. However, they overlook some beneficial dependency information given the rich inner structure of the TDT and FTB labels discussed in Sect. 2. Therefore, we follow Silfverberg et al. (2014) and utilize an expanded feature set which aims to capture these dependencies. For example, consider the word form *kissat* (*cats*) where the suffix *-t* denotes plural number. Then, given the feature extraction scheme of Silfverberg et al. (2014), instead of associating the suffix *-t* solely with a compound label (Noun, nominative, plural), we also relate it with the sub-label Plural. This is because one can exploit the suffix *-t* to predict the plural number also in words such as *vihreät* (*plural of green*) with an analysis (Adjective, nominative, plural).

In addition to associating the input to sub-labels as described above, the expanded feature set exploits transitional behavior of the sub-labels. For example, consider the sentence fragment *kissat juovat* (*cats drink*) where the words *kissat* and *juovat* have compound analyses (Noun, nominative, plural) and (Verb, 3rd person, plural, present tense, active), respectively. Then, instead of merely modeling the transitional dependency between the compound labels, we also model the congruence, that is, both analyses need to contain the sub-label denoting plural number.

3.1.3 Estimation from data

In this section, we describe the averaged perceptron parameter estimation procedure implemented in the FinnPos system.

Averaged perceptron learning

The perceptron learning algorithm operates by iteratively searching for the highest scoring label sequence for a training instance x and updating the model parameters in case of an incorrect search result. From implementation perspective, the exact search is performed using the standard Viterbi algorithm.

Inconveniently, however, perceptron learning utilizing exact Viterbi search is impractically slow in presence of large label sets. Therefore, in order to speed up the estimation, we implement the perceptron algorithm utilizing *beam search* using *minimum divergence beams* following Pal et al. (2006).[2] Finally, the parameter averaging technique (Freund and Schapire 1999; Collins 2002) provides a simple, hyper-parameterless means of model regularization.

Model cascade

In a recent work, Müller et al. (2013) presented an approximative high-order CRF estimation technique utilizing a cascade of CRF models of increasing orders. In a general cascade system for structured prediction, one learns a series of increasingly complex models by restricting the search space of each model using the predictions of the less complex models (Weiss and Taskar 2010). Müller et al. (2013) implement this approach for CRFs using a coarse-to-fine decoding technique (Charniak and Johnson 2005; Rush and Petrov 2012) and show large savings in the computational cost of maximum likelihood training.

In the FinnPos system, we implement another cascading variant by applying a series of two models, an orthography-based label guesser and a conventional nth-order perceptron classifier. In this approach, the idea is simply to utilize the minimalistic, orthography-based label guesser to narrow down the label search space. This roughly corresponds to the zero-order pruning performed by Müller et al. (2013). In order to apply the cascade, we first learn the label guesser from the training data. The guesser ranks morphological tags according to their probability for any given word form. We then use the guesser to limit the candidate label set for each word x_i in sentence x and, subsequently, use beam search to find the highest scoring label sequence among the limited candidate sequences. Parameter updates are performed in a standard manner.

The implemented label guesser is based on the lexical model for OOV words used by Brants (2000). It assigns a probability $p(y|x)$ for any label $y \in \mathcal{Y}$ and an arbitrary word x based on the suffixes of x. Appealingly, the guesser can be trained and applied in mere seconds even when using large data sets.

We use the label guesser to extract the minimal set of highest ranking label guesses y_i whose combined probability mass $\sum_{i=0}^{n} p(y_i|x)$ exceeds a threshold $\kappa \in [0,\ 1]$. The threshold is considered a hyper-parameter of the learning procedure, which is tuned on a held-out development set. Essentially, if one employs too small a κ, the model will underfit the training data, while increasing the threshold results in increasingly accurate approximations of the original perceptron learning problem.

3.1.4 Decoding

Subsequent to parameter estimation using the learning algorithms discussed in Sect. 3.1.3, the resulting perceptron classifier can be applied to any given word sequence. In this decoding stage, the model assigns the highest scoring label

[2] In addition to applying standard beam search and parameter updates, we experimented with the maximum violation and early updates of Huang et al. (2012) but obtained no improvements in model accuracy.

sequence to a given word sequence. The search is performed using beam search with the minimum divergence beams following Pal et al. (2006). In addition, since model estimation is performed utilizing a label guesser, the label guesser is also applied during decoding with the same threshold value κ. Lastly, if a morphological analyzer is available during decoding, the labels of test instances are restricted according to the output of the analyzer.

3.2 Lemmatizer

In order to lemmatize words unknown to the OMorFi analyzer, we follow Chrupala et al. (2008) and treat the lemmatization problem as a classification task, in which each class corresponds to a *suffix edit script*. For example, consider $[ies \rightarrow y]$, which removes a suffix "-ies" from the end of a word form, such as the English word form "beauties", and replaces it with another suffix "-y", thus producing the lemma "beauty". While Chrupala et al. (2008) use rather general edit scripts which can additionally modify prefixes and infixes of the word, we rely on the suffix-based approach because Finnish words mostly inflect at the end. The task of the lemmatizer is then to find the most appropriate edit script based on features extracted from the word form, its morphological label and its context. The script is chosen among *minimal edit scripts*, where the removed suffix is as short as possible (Chrupala et al. 2008).

3.3 Implementation details

This section describes low-level details involved in the implementation of the tagging and lemmatization methods discussed above in Sects. 3.1 and 3.2, respectively. The covered issues include hyper-parameter tuning, initialization procedures, and software.

Feature Extraction.

The default feature extraction follows the presentation in Sect. 3. However, users can freely define their own feature sets.[3] The toolkit implements label and sub-label transitions discussed in Sect. 3.1.2 up to second order. The transition orders are optimized based on the development set. We use prefixes and suffixes of words up to length 10 ($\delta_{affix} = 10$).

3.3.1 Averaged Perceptron and Label Guesser

The perceptron algorithm initializes model parameters with a zero vector. In order to reduce overfitting, we apply the parameter averaging approach following Collins (2002) and an early stopping criterion based on the held-out development set. In early stopping, we apply the averaged parameters to the development set after each pass over the training data and terminate training in case the accuracy has not improved during the previous three passes. Subsequently, we apply the best

[3] See documentation at https://github.com/mpsilfve/FinnPos/wiki.

performing parameter setting to the test instances. The label guesser is trained using all words in the training data utilizing all word suffixes up to length 10.

3.3.2 Lemmatizer

Given lemmatized training data, we first extract all minimal suffix edit scripts. We then train an averaged perceptron classifier (Freund and Schapire 1999) to disambiguate between all applicable suffix edit scripts for each word form in the training data. The classifier uses the following feature set:

1. The word form w.
2. Suffixes of w up to length 10.
3. Infixes (w_{n-4}, w_{n-3}), (w_{n-3}, w_{n-2}) and (w_{n-2}, w_{n-1}) of the word form $w = (w_1, \ldots, w_n)$.
4. The morphological label of w as well as its sub-labels.
5. If the word form w contains (one or more) capital letters, hyphens, dashes, or digits.

Additionally, we use combination features where each feature is combined with the morphological label of w and its sub-labels. Overfitting to training data is controlled using parameter averaging and early stopping based on the development data.

3.3.3 Implementation

To guarantee efficient estimation and inference, FinnPos is implemented in C++. In order to facilitate compilation and avoid clashes between library versions, we eliminated most dependencies on external software and libraries. Currently, the only required external utility is the lookup program for the OMorFi morphological analyzer distributed with the HFST library (Lindén et al. 2011).

4 Experiments

In this section, we present an empirical evaluation of the FinnPos system on two Finnish treebanks. The evaluation considers tagging and lemmatization accuracy and computational efficiency of learning and decoding. For comparison, we provide results using three reference toolkits.

4.1 Data

The experiments are conducted on the Turku Dependecy Treebank (Haverinen et al. 2009, 2014) and FinnTreeBank (Voutilainen 2011) described in Sect. 2. The treebanks do not have default partitions to training and test sets. Therefore, from each ten consecutive sentences, we assign the 9th and 10th to the development sest and the test sets, respectively. The remaining sentences are assigned to the training sets. Statistics for the data splits are given in Table 3.

Table 3 Sizes of the training, development and test sets for FTB and TDT

	TDT	FTB
Train	10,858 sent. (145,775 tok.)	15,297 sent. (129,374 tok.)
Dev	1357 sent. (18,060 tok.)	1912 sent. (16,579 tok.)
Test	1357 sent. (19,283 tok.)	1912 sent. (16,075 tok.)
OOV in test	21.9 %	22.1%

The last row indicates the amount of tokens in the test set that are not found in the train set

Table 4 The main POS distributions of all and out-of-vocabulary (OOV) words for the test set of Turku Dependency Treebank

Label	Example	All words		OOV words	
		Absolute Frequency	Relative Frequency (%)	Absolute Frequency	Relative Frequency (%)
Noun	talo (a house)	6565	34.0	2656	62.9
Verb	istua (to sit)	3810	19.8	872	20.6
Punctuation	. " ,	2897	15.0	0	0.0
Adverb	nopeasti (quickly)	1407	7.3	79	1.9
Adjective	hidas (slow)	1243	6.4	447	10.6
Pronoun	sinä (you)	1241	6.4	36	0.9
Conjunction	kun (when)	1096	5.7	2	0.0
Numeral	kolme (three)	652	3.4	73	1.7
Adposition	alla (under)	285	1.5	6	0.1
Foreign	live (live)	37	0.2	29	0.7
Symbol	:D	32	0.2	19	0.4
Interjection	nam (yum)	18	0.1	5	0.1

Tables 4 and 5 show the distributions of main POS classes for the test sets of TDT and FTB, respectively. Although the morphological labeling schemes in both FTB and TDT follow the labeling scheme of the OMorFi morphological analyzer, they are based on different versions of OMorFi. Therefore, the treebanks have differing main POS inventories. For example, the class Particle in FTB overlaps with the classes Conjunction and Adverb in TDT.

The encoding of nouns in FTB and TDT differs with regard to coordinated compounds. In Finnish, a coordination of two compound words which share an identical part can be written in an abbreviated manner. For example, *isotuloiset ja pienituloiset* (*people with high income and people with low income*) can be abbreviated as *iso- ja pienituloiset* because the coordinated compounds share the final part *-tuloiset*. FTB denotes the compound prefix *iso-* by a separate main POS Truncated. In contrast, TDT labels these prefixes as regular nouns or adjectives.

Both FTB and TDT group common and proper nouns under the main POS Noun. The distinction is, however, denoted by an additional subcategory label.

Table 5 The main POS distributions of all and out-of-vocabulary (OOV) words for the test set of FinnTreeBank

Label	Example	All words		OOV words	
		Absolute Frequency	Relative Frequency (%)	Absolute Frequency	Relative Frequency (%)
Noun	talk (a house)	4354	27.1	2079	58.6
Verb	istua (to talk)	3831	23.8	755	21.3
Punctuation	. ” ,	2302	14.3	0	0.0
Particle	näin (thus)	1502	9.3	23	0.6
Pronoun	sinä (you)	1437	8.9	37	1.0
Adverb	nopeasti (quickly)	1040	6.5	112	3.2
Adjective	hidas (slow)	1033	6.4	431	12.1
Numeral	kolme (three)	278	1.7	74	2.1
Adposition	alla (under)	273	1.7	16	0.5
Unknown	live (live)	16	0.1	14	0.4
Truncated	iso- (big)	9	0.1	8	0.2

4.2 Reference systems

This section summarizes the reference systems, namely, Morfette (Chrupala et al. 2008), MarMot (Müller et al. 2013), and HunPos (Halácsy et al. 2007).

Morfette. Morfette is a toolkit for learning a morphological tagging and lemmatization model from annotated training data.[4] Given a corpus of sentences annotated with lemmas and morphological labels, and optionally a morphological analyzer, Morfette learns to assign analyses for new sentences. The Morfette tagging model is based on the averaged perceptron classifier. Meanwhile, lemmatization is handled as a classification task, in which each lemmatization class corresponds to a set of string edit operations required to transform the inflected word form into the corresponding lemma. This general approach is adopted by FinnPos.

MarMot. MarMot is a CRF-based morphological tagging toolkit.[5] Given a corpus of sentences annotated with morphological labels, and optionally a morphological analyzer, MarMot learns to assign morphological tags for new sentences. The model estimation of MarMot is based on the maximum likelihood criterion utilizing a pruning approach which enables efficient learning of high-order models. In contrast to FinnPos and Morfette systems, MarMot is solely a morphological tagging toolkit and does not perform lemmatization.

HunPos. HunPos is an improved, open-source implementation of the morphological TnT tagger of Brants (2000).[6] Given a corpus of sentences annotated with morphological labels, and optionally a morphological analyzer, HunPos learns to

[4] Available at https://sites.google.com/site/morfetteweb/.

[5] Available at https://code.google.com/p/cistern/wiki/marmot.

[6] Available at http://code.google.com/p/hunpos/.

assign morphological tags for new sentences. Similarly to MarMot, HunPos is solely a morphological tagging toolkit and does not perform lemmatization. The HunPos tagger is based on the generative HMM framework which makes it sensitive to rich feature sets compared to the discriminatively trained perceptron classifier and CRFs. On the other hand, due to the generative estimation procedure and simple feature sets, the system is extremely fast to both train and apply. While the HunPos system was originally designed for morphological tagging of Hungarian, it is a natural choice for a Finnish morphological tagger due to the relatedness of Hungarian and Finnish languages: Hungarian and Finnish are both agglutinative and morphologically rich languages belonging to the Finno-Ugric family.

4.3 Evaluation

Test performances in tagging and lemmatization (when applicable) are evaluated using *per-token* accuracies. These accuracies are reported separately for all words and words not seen in the training data. We establish statistical significance (with confidence level 0.95) using the standard 2-sided Wilcoxon signed-rank test performed on ten randomly divided, non-overlapping subsets of the complete test sets.

4.4 Hardware

The experiments are run on a desktop computer (Intel Core i5-4300U with 1.90 GHz and 16 GB of memory).

4.5 Results

Obtained tagging and lemmatization accuracies, training times, and decoding speeds for TDT and FTB are presented in Tables 6 and 7, respectively. In what follows, we will compare the FinnPos system individually with the reference systems.

FinnPos versus Morfette We begin by comparing FinnPos and Morfette, both of which perform morphological tagging and lemmatization. The FinnPos system outperforms the Morfette with respect to both tagging and lemmatization accuracy. The differences in accuracies are statistically significant. Furthermore, compared to FinnPos, the training time of Morfette is substantially higher and decoding speed substantially lower.

FinnPos versus MarMot The FinnPos system outperforms MarMot with respect to tagging accuracy. However, the differences in accuracies are not statistically significant. Compared to FinnPos, the training time of MarMot is substantially higher and decoding speed substantially lower. Finally, MarMot does not perform lemmatization.

FinnPos versus HunPos The training time of the HunPos system is substantially lower compared to FinnPos or any other system. While faster to estimate and apply, however, the tagging accuracy of HunPos is significantly lower compared to FinnPos on both data sets. The HunPos system does not perform lemmatization.

Table 6 Results for Turku Dependency Treebank

Toolkit	Tag acc.		Lemma acc.		Train. time		dec. speed (tok/s)
	All	OOV	All	OOV	Tagger	Lemmatizer	
HunPos	91.64	76.07	–	–	2 s	–	101,000
MarMot	96.29	91.04	–	–	38 min	–	1000
Morfette	93.91	82.19	89.33	72.04	203 min	16 min	40
FinnPos	**96.31**	**91.64**	**93.29**	**84.28**	4 min	5 min	16,000

The bold values indicate the accuracy of the best performing system

Table 7 Results for FinnTreeBank

Toolkit	Tag acc.		Lemma acc.		Train. time		dec. speed (tok/s)
	All	OOV	All	OOV	Tagger	Lemmatizer	
HunPos	93.65	82.55	–	–	2 s	–	141,000
MarMot	96.21	91.46	–	–	24 min	–	1000
Morfette	95.03	86.81	95.66	83.12	128 min	8 min	60
FinnPos	**96.23**	**92.34**	**97.49**	**92.85**	3 min	3 min	18,000

The bold values indicate the accuracy of the best performing system

4.6 Error analysis

In this section, we present and discuss the distribution of the errors yielded by the FinnPos system. In particular, we examine how the errors are distributed across the main POS classes. In addition, we examine individual error types, that is, which categories are most often confused for one another.

First, consider Tables 8 and 9 which contain the errors distributions for TDT and FTB, respectively. For both data sets, the majority of errors take place in the noun and verb categories. This is expected as these categories are most frequent in the test sets and, as shown in Tables 4 and 5, and contain the most OOV word forms.

Second, consider Tables 10 and 11 which contain confusion matrices of errors for TDT and FTB, respectively. Due to space constraints, the matrices include 25 most prominent confusion pairs. For both data sets, the majority of errors take place when a noun is confused with a noun or a verb is confused with a verb, that is, the tagger yields the correct main POS class but an incorrect detailed morphological label. For example, consider the noun phrase *kiveä ja terästä oleva monumentti* (*a monument made of stone and steel*).[7] The word form *terästä* could be the partitive form of the noun *teräs* (steel) or the elative form of the noun *terä* (a blade). From a syntactical point of view, both interpretations are possible. From a semantical point of view, however, only the partitive interpretation is valid.

[7] The example is taken from FinnTreeBank.

Table 8 Error distribution over main POS classes for Turku Dependency Treebank

Main POS	All words		OOV words	
	Absolute Frequency	Relative Frequency (%)	Absolute Frequency	Relative Frequency (%)
Noun	271	38.1	182	53.5
Verb	205	28.8	61	17.9
Adjective	65	9.1	33	9.7
Pronoun	48	6.8	17	5.0
Adverb	45	6.3	11	3.2
Foreign	23	3.2	21	6.2
Numeral	15	2.1	4	1.2
Adposition	15	2.1	2	0.6
Conjunction	13	1.8	1	0.3
Symbol	7	1.0	7	2.1
Interjection	4	0.6	1	0.3

Table 9 Error distribution over main POS classes for FinnTreeBank

Main POS	All words		OOV words	
	Absolute Frequency	Relative Frequency (%)	Absolute Frequency	Relative Frequency (%)
Noun	184	30.4	127	49.0
Verb	155	25.6	54	20.8
Adverb	71	11.7	11	4.2
Particle	47	7.8	4	1.5
Pronoun	46	7.6	3	1.2
Adjective	40	6.6	27	10.4
Numeral	24	4.0	11	4.2
Adposition	17	2.8	3	1.2
Unknown	12	2.0	11	4.2
Truncated	8	1.3	8	3.1
Punctuation	2	0.3	–	–

4.7 Discussion

Compared to the reference toolkits, the FinnPos system provides the highest accuracies with respect to tagging and lemmatization accuracy. In addition, the system is computationally more efficient to train and apply compared to the MarMot and Morfette systems which also utilize discriminative learning. As discussed in Sect. 2, the TDT and FTB corpora differ somewhat in the included text domains as well as the labeling schemes. However, these differences appear to have a minor effect on the tagging and lemmatization accuracy of the FinnPos system.

Table 10 The confusion matrix of errors for Turku Dependency Treebank with relative error frequencies

	Noun	Verb	Adjective	Pronoun	Adverb	OTHER
Noun	26.0	2.7	3.1	0.7	1.5	4.1
Verb	5.1	20.5	2.3	0.4	0.1	0.4
Adjective	2.3	2.4	3.0	0.0	1.4	0.1
Pronoun	1.5	0.1	0.0	2.4	1.3	1.4
Adverb	1.0	0.1	0.6	1.1	0.3	3.2
OTHER	4.2	1.0	0.1	0.1	2.1	3.2

For example, labeling a noun as a verb comprises 2.7 percentages of all errors, whereas labeling a verb as a noun comprises 5.1 percentages of all errors. For the five main POS classes with most labeling errors, results are shown separately. The class OTHER comprises all remaining main POS classes

Table 11 The confusion matrix of errors for FinnTreeBank with relative error frequencies

	Noun	Verb	Adverb	Particle	Pronoun	OTHER
Noun	17.7	1.7	3.0	1.5	0.0	6.6
Verb	2.5	17.2	0.3	1.2	0.5	4.0
Adverb	1.2	0.0	1.7	3.5	2.0	3.5
Particle	1.0	0.5	1.3	4.1	0.7	0.2
Pronoun	0.2	0.0	2.5	0.3	3.6	1.0
OTHER	5.6	2.8	3.0	0.3	0.7	4.6

According to the error analysis in Sect. 4.6, while the main POS label is often correct, the detailed morphological information is more difficult to infer. The analysis shows that substantial improvement in tagging accuracy would require improved inference of the detailed morphological information for nouns and verbs specifically. This, however, is a difficult task because the immediate syntactical context often does not provide adequate clues for disambiguation. The choice between different detailed labels is often lexically and semantically conditioned which makes it particularly difficult for OOV words.

5 Conclusions

We presented FinnPos, an open-source morphological tagging and lemmatization toolkit for Finnish. The toolkit is readily applicable for tagging and lemmatization of running text with models learned from the recently published Finnish Turku Dependency Treebank and FinnTreeBank. The performed empirical evaluation showed that FinnPos performs favorably to several reference systems (MarMot, Morfette, HunPos) in terms of tagging and lemmatization accuracy, as well as computational efficiency of learning new models and assigning analyses to novel sentences.

The FinnPos system should be readily applicable for learning taggers for languages closely related to Finnish, such as Hungary and Estonian. On the other hand, the default feature extraction scheme may also perform well on other morphologically rich European languages, such as Czech and Romanian. Therefore, in future work, the toolkit could be evaluated empirically on multiple languages.

References

Bohnet, B., Nivre, J., Boguslavsky, I., Ginter, R. F. F., & Hajič, J. (2013). Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics, 1,* 415–428.

Brants, T. (2000). TnT: A statistical part-of-speech tagger. In *Proceedings of the 6th conference on applied natural language processing (ANLP 2000)* (pp. 224–231). Washington, USA: Seattle.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd annual meeting on association for computational linguistics (ACL 2005)* (pp. 173–180). Ann Arbor: Michigan, USA.

Chrupala, G., Dinu, G., & van Genabith, J. (2008). Learning morphology with Morfette. In *Proceedings of the 6th international conference on language resources and evaluation (LREC 2008)* (pp. 2362–2367). Morocco: Marrakech.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)* (Vol. 10, pp. 1–8). Philadelphia, Pennsylvania, USA.

Freund, Y., & Schapire, R. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*(3), 277–296.

Hakulinen, A., Korhonen, R., Vilkuna, M., & Koivisto, V. (2004). Iso suomen kielioppi. Suomalaisen kirjallisuuden seura, http://scripta.kotus.fi/visk.

Halácsy, P., Kornai, A., & Oravecz, C. (2007). HunPos: An open source trigram tagger. In *Proceedings of the 45th annual meeting of the association of computational linguistics (ACL 2007)* (pp. 209–212). Prague: Czech Republic.

Haverinen, K., Ginter, F., Laippala, V., Viljanen, T., & Salakoski, T. (2009). Dependency annotation of Wikipedia: First steps towards a Finnish treebank. In *The 8th international workshop on treebanks and linguistic theories (TLT 2009)* (pp. 95–105). Milan: Italy.

Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., Ojala, S., Salakoski, T., & Ginter, F. (2014). Building the essential resources for Finnish: The Turku Dependency Treebank. *Language Resources and Evaluation, 48*(3), 493–531.

Huang, L., Fayong, S., & Guo, Y. (2012). Structured perceptron with inexact search. In *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies (NAACL HLT 2012)* (pp. 142–151). Canada: Montreal.

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th conference on computational linguistics (COLING 1990)* (pp. 168–173). Finland: Helsinki.

Lindén, K., Axelson, E., Hardwick, S., Pirinen, T., & Silfverberg, M. (2011). HFST—Framework for compiling and applying morphologies. *Systems and Frameworks for Computational Morphology (SFCM 2011)* (pp. 67–85). Switzerland: Zurich.

Müller, T., Schmid, H., & Schütze, H. (2013). Efficient higher-order CRFs for morphological tagging. In *Proceedings of 2013 empirical methods in natural language processing (EMNLP 2013)* (pp. 322–332). Washington, USA: Seattle.

Pal, C., Sutton, C., & McCallum, A. (2006). Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Internation conference on acoustics, speech and signal processing (ICASP 2006)* (Vol. 5, pp. 581–584). Toulouse, France.

Pirinen, T. (2008). *Automatic finite state morphological analysis of Finnish language using open source resources (in Finnish).* Master's thesis, University of Helsinki.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1996 conference on empirical methods in natural language processing (EMNLP 1996)* (Vol.1, pp. 133–142). New Brunswick, New Jersey, USA.

Rush, A. M., & Petrov, S. (2012). Vine pruning for efficient multi-pass dependency parsing. In *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies (NAACL HLT 2012)* (pp. 498–507). Canada: Montreal.

Silfverberg, M., & Linden, K. (2011). Combining statistical models for POS tagging using finite-state calculus. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)* (pp. 183–190). Latvia: Riga.

Silfverberg, M., Ruokolainen, T., Lindén, K., & Kurimo, M. (2014). Part-of-speech tagging using conditional random fields: Exploiting sub-label dependencies for improved accuracy. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics (ACL 2014)* (pp. 259–264). Maryland: Baltimore.

Sutton, C., & McCallum, A. (2011). An introduction to conditional random fields. *Machine Learning*, *4*(4), 267–373.

Voutilainen, A. (2011). FinnTreeBank: Creating a research resource and service for language researchers with Constraint Grammar. In *Proceedings of the NODALIDA 2011 workshop constraint grammar applications* (pp. 41–49). Latvia: Riga.

Weiss, D., & Taskar, B. (2010). Structured prediction cascades. In *International conference on artificial intelligence and statistics (AISTATS 2010)* (pp. 916–923). Italy: Sardinia.