Part-of-speech tagging of Swedish texts in the neural era

Yvonne Adesam and **Aleksandrs Berdicevskis**
Språkbanken Text
Department of Swedish
University of Gothenburg
yvonne.adesam@gu.se, aleksandrs.berdicevskis@gu.se

Abstract

We train and test five open-source taggers, which use different methods, on three Swedish corpora, which are of comparable size but use different tagsets. The KB-Bert tagger achieves the highest accuracy for part-of-speech and morphological tagging, while being fast enough for practical use. We also compare the performance across tagsets and across different genres. We perform manual error analysis and perform a statistical analysis of factors which affect how difficult specific tags are. Finally, we test ensemble methods, showing that a small (but not significant) improvement over the best-performing tagger can be achieved.

1 Introduction

The standard approach to automatic part-of-speech tagging for Swedish has been using the Hunpos tagger (Halácsy et al., 2007), trained by Megyesi (2009) on the Stockholm-Umeå corpus (Ejerhed et al., 1992). Just over a decade later neural methods have reshaped the NLP landscape, and it is time to re-evaluate which taggers are most accurate and effective for Swedish text.

In this paper we explore part-of-speech and morphological tagging for Swedish text. The primary purpose is to see which tagger or taggers to include in the open annotation pipeline Sparv[1] (Borin et al., 2016) for tagging the multi-billion token corpora of Språkbanken Text, available through Korp[2] (Borin et al., 2012). We therefore train and test a set of part-of-speech taggers, which rely on different methods, on a set of corpora of comparable size, with different part-of-speech annotation models. We apply a 5-fold training and evaluation regime.

In Section 2 we describe the corpora, and in Section 3 the taggers and models. We evaluate the taggers along a number of dimensions in Section 4, including the potential for using ensemble methods, and discuss the results in Section 5.

2 Data

2.1 Corpora and tagsets

Corpora and treebanks have a long history in Sweden; the first large annotated treebank, Talbanken, was compiled in the mid 1970s (Teleman, 1974). For several decades, the Stockholm-Umeå corpus (SUC, Ejerhed et al., 1992) has been the main resource for training part-of-speech taggers.

In this paper, however, we use three other corpora: Talbanken-SBX, Talbanken-UD, and Eukalyptus. The primary reason for using these three resources is that they are annotated with different tagsets, which allows us to compare results between tagsets. Talbanken-SBX follows the same annotation model as SUC. Talbanken-UD follows the Swedish version of the Universal Dependencies (UD) framework (Nivre et al., 2016; Nivre, 2014). The UD project develops a cross-linguistic annotation framework and resources annotated with it for a large number of languages. In contrast, the Eukalyptus treebank (Adesam et al., 2015) was developed specifically for Swedish to be "in line with the currently standard view on Swedish grammar" (Adesam and Bouma, 2019, p. 7). We also exclude SUC because these three resources are of comparable size – close to 100,000 tokens and with a type-token ratio of around 0.17. SUC is much larger, and would have to be scaled down to be comparable.

We briefly describe the corpora below. For consistency, we use the same terms to describe the annotation in the corpora: POS for coarse-

[1] https://spraakbanken.gu.se/sparv

[2] https://spraakbanken.gu.se/korp

	TB-SBX	TB-UD	Euk
Tokens	96,346	96,858	99,909
Types	16,242	16,305	17,237
POS-tags	25	16	13
MSD-tags	130	213	117

Table 1: Statistics for the corpora used in the tagging experiments; Talbanken-SBX, Talbanken-UD, and Eukalyptus. Tag counts are used tags, not potential tags.

grained part-of-speech level tags and MSD for finer-grained morphosyntactic descriptions (*features* in the UD parlance).

The two Talbanken corpora originate from a subset (the professional prose section) (Nivre et al., 2006) of the original Talbanken (Teleman, 1974), which was converted to the SUC tagset (Ejerhed et al., 1992) for the Swedish Treebank (Nivre and Megyesi, 2007)[3]. The morphological annotation was manually checked and revised. Both Talbanken-SBX and Talbanken-UD are based on the output of this conversion.

Talbanken-SBX[4] has the converted SUC tags, and is the result of some minor corrections made later at Språkbanken Text. Among our three corpora, the SUC tagset is the largest set at the POS-level (see Table 1). It has a very fine-grained set of tags for determiners, pronouns, adverbs, and punctuation symbols. There are also separate tags for infinitival markers, participles, verb particles, and ordinals.

Talbanken-UD[5] is the result of an independent conversion of the same corpus to UD. The texts themselves were cleaned during this conversion, some sentences that had been lost during the initial conversion were recovered, and sentence segmentation and the order of texts was changed. Thus, Talbanken-UD and Talbanken-SBX are not strictly parallel. The conversion to UD has partly been manually checked and revised. We use version 2.7.

The number of POS-tags in the UD tagset is quite small, but together with MSD-tags the tagset is the largest among our corpora (Table 1). The tagset does not have separate categories for the infinitival marker, ordinals, or participles. It also does not mark foreign words as a category, but instead treats this as a feature in the morphological description. In contrast to the other tagsets, it does, however, mark auxiliaries separately.

Eukalyptus[6] contains texts of five different types, including Wikipedia and blog texts, which makes this data the most recent and allows us to compare different genres. The tagset loosely builds upon the SUC tagset. The treebank is currently in an early version, and although tagging has been checked, there are still some known errors, such as inconsistencies in noun gender. This tagset is the smallest one, both at POS- and MSD-levels (Table 1). The tagset does not, for example, distinguish determiners, infinitival markers, participles, particles, or ordinals as separate categories.

2.2 Preprocessing and data splits

We pre-processed all corpora in a similar manner. For all corpora, spaces within tokens, if present, were replaced with underscore, since some taggers do not allow spaces in the input. We divided all three datasets into five folds for cross-validation. In the case of Eukalyptus, the treebank is shipped in five different files, one for each text type, which were used as is. In the case of Talbanken, we split the data into five consecutive splits, i.e. putting the first fifth of the data into the first split, the second fifth in the second, etc. We would have preferred to divide the data according to text types or documents, but this is not easily retrievable for all the data. Using consecutive splits rather than random splits or splits where the first sentence is put in the first split, the second sentence in the second split, etc, means that the data splits are more distinct than with random splits (see the discussion in e.g. Gorman and Bedrick, 2019; Søgaard et al., 2020). This means that the same text is not divided over all splits, although possibly into two splits.

One of the five folds (20%) is always used a test set. Some of the taggers we investigated do use a separate validation (dev) set, some do not (see Table 2). For the latter ones, we merge all four remaining folds into a training set (80%). For the former ones, we first merge the four folds and then

[3] https://cl.lingfil.uu.se/~nivre/swedish_treebank/

[4] https://spraakbanken.gu.se/en/resources/talbanken

[5] https://universaldependencies.org/treebanks/sv_talbanken/index.html

[6] https://spraakbanken.gu.se/en/resources/eukalyptus

randomly (not consecutively) split them into train and dev in the proportion 3:1 (60% of the total data for training and and 20% for validation). We consider this solution to be more fair to the "dev-less" taggers than using the same training sets throughout and then adding dev for some taggers, but not for others.

3 Taggers

We have selected five open-source taggers. Our goal was to sample taggers that use different methods, are (or were at some point) known to have high performance and either can be easily incorporated into our annotation pipeline Sparv or already are (as Hunpos and Stanza). This last consideration steers the selection to a large extent (Stanza, for instance, has an important advantage of being a convenient pipeline that achieves high performance on other tasks, such as dependency parsing).

We also wanted to compare taggers that were state-of-the-art in the "pre-neural" era[7] with the current ones. The key properties of the taggers are summarized in Table 2. Note that the classification in the "Key method" column is of course very crude (Flair, for instance, can be labelled as both neural and CRF).

As can be seen from the table, different taggers use different kinds of additional information. Hunpos does not take any further input. For Marmot, we plug in Saldo-Morphology (Borin et al., 2013), a morphological dictionary of 1M words with a tagset that is similar (but not equivalent) to the SUC tagset. From previous experiments we know that using Saldo gives Marmot a boost when it is applied to texts tagged with the SUC tagset (i.e. TalbankenSBX in our case). We assume it can also boost performance on Eukalyptus, since the tagsets are similar, but we do not expect a boost for UD. For Stanza, we use word2vec embeddings[8] trained on the CONLL17 corpus (Zeman et al., 2017), which was built using the CommonCrawl data and contains approximately 3 billion words for Swedish. One of the main ideas of Flair is to combine various types of embeddings; the best combination we were able to find was that of the CONLL17 word2vec and Flair's own embeddings (trained on Wikipedia/OPUS[9], size is not reported). For KB-Bert[10], we used the `bert-base-swedish-cased` model, trained by the Datalab of the National Library of Sweden (KB) on 3.5 billion words from the library collection. The collection contains predominantly (85%) newspaper texts, but also official reports from authorities, books, magazines, social media and Wikipedia. The training and tagging itself was done as in (Malmsten et al., 2020), using the `run_ner.py` script from the Huggingface framework[11]. For Stanza and Flair, we experimented with using different classic and contextualized embeddings, for instance, word2vec trained on a press corpus (Fallgren et al., 2016) or Bert instead of Flair's own embeddings, but the results were always slightly worse than those we report.

4 Evaluation

We evaluate the taggers on the treebanks along several dimensions. In the following we report tagger speed and accuracy. We also explore unseen words, specific tags that seem more difficult to get right, as well as an ensemble approach.

4.1 Speed

We trained the neural taggers on GPU (on CPU the training time is prohibitively long) and the non-neural ones on CPU. This means the time measurements are not directly comparable, and we thus do not report detailed quantitative results, but the qualitative picture is very clear. For Hunpos, the training on one fold takes about a second, so does tagging. For Marmot, training takes about 1.5 minutes, tagging about 10 seconds. For Stanza, training takes about 2 hours, tagging about 8 seconds. For Flair, training takes about 6 hours, tagging about 5 seconds. KB-Bert, however, breaks the pattern "the better the slower": training takes about 3 minutes, tagging takes about 5 seconds. Note that for the neural taggers the tagging time

[7] An anonymous reviewer notes that the best label for the current era is not "neural", but "post-neural" or "language-model" era.

[8] `http://vectors.nlpl.eu/repository`

[9] `https://github.com/flairNLP/flair/blob/master/resources/docs/embeddings/FLAIR_EMBEDDINGS.md`

[10] The script crashes if the dev set contains previously unseen tags. To solve this, we replace all such tags with the tag for adverb (AB for SBX and Eukalyptus, ADV for UD) when training Bert. This can potentially affect the results, but the number of such tags is always small (varying from 0 to 10 across various folds), which should only give a negligible bias against KB-Bert.

[11] `https://github.com/huggingface/transformers/blob/master/examples/token-classification`

		Embeddings				
Name	Key method	Token	Type	Dictionary	Dev	References
KB-Bert	Neural	KB-BERT	-	-	Yes	Malmsten et al. (2020); Wolf et al. (2020)
Flair	Neural	Flair	Word2vec	-	Yes	Akbik et al. (2019)
Stanza	Neural	-	Word2vec	-	Yes	Qi et al. (2020)
Marmot	CRF	-	-	SALDO	No	Mueller et al. (2013)
Hunpos	HMM	-	-	-	No	Halácsy et al. (2007)

Table 2: Basic info about the taggers. HMM = hidden Markov models, CRF = conditional random fields, Dev = whether the tagger uses a development set. Type embeddings = "classic" ("static") embeddings, token = "contextualized" ("dynamic").

excludes the time necessary to load models, embeddings and all necessary modules. If this is taken into account, the tagging time becomes considerably longer (for KB-Bert, for instance, about 30 seconds).

4.2 Overall tagging quality

Table 3 shows the accuracy (macroaverage over 5 folds) for the full POS+MSD label. It shows that KB-Bert achieves the best results, and that the Talbanken-SBX corpus is easiest to tag, while Eukalyptus has lower results. It is not surprising that the newer neural models perform the best, while the older models achieve lower scores. To test whether differences between the taggers are significant, we rank them by performance and then do pairwise comparisons of adjacent taggers (KB-Bert and Flair, Flair and Stanza etc.) by running paired two-tailed t-tests on 15 (3x5) datapoints. We apply the same procedure to the sentence-level accuracy (Table 5) and to accuracy on unseen words (Table 7). All the differences are significant ($p < 0.05$ level) and have non-negligible effect size (Cohen's $d > 0.2$). The results remain significant after applying the Bonferroni correction for multiple comparisons.

One may wonder if Eukalyptus has more difficult distinctions, or is more inconsistently annotated. However, it should be noted that the variation between splits is much larger for Eukalyptus than for the other two corpora. If we disregard testing on the blog part (although we still include it for training) the 4-fold macro average is more similar to the Talbanken-UD results, although still lower. However, the standard deviation (SD) is also still higher than for the other two corpora. The reason for this may be the distinctiveness of text types or genres of the Eukalyptus parts.

To check this, we also ran KB-Bert on randomized versions of the three corpora, where sentences are randomly assigned to folds. This means that the differences are evened out between folds and that the test data is more similar to the training data. The results are shown in Table 4. As we can see, the results between the three corpora are more similar than for the consecutive splits (with Eukalyptus even getting better results than Talbanken-UD). SD between folds is very low, except for Talbanken-UD. However, since the random assignment of sentences to splits makes tagging easier, all results reported in this paper, except for in Table 4, are based on the consecutive splits, not the random splits.

In Table 5 we look at sentence-level accuracy, that is the amount of sentences where all words have the correct tag. The pattern is the same as for the token-level results in Table 3 regarding which tagger performs the best, but the distance between Bert and the other taggers is even greater. However, the differences between folds are also greater.

4.3 Unseen words

Since training data can never contain all potential words or word-tag combinations, how well a tagger does on words previously unseen in the training data (OOV) is important, and often varies between different methods.

In Table 6 we show the numbers of unseen words, averaged over the five folds of each corpus. It is clear that the different folds for Talbanken-SBX and Talbanken-UD are quite similar, while there are larger differences between the folds of Eukalyptus. There, the Wikipedia part has the largest number of OOV word forms.

Table 7 shows tagging results for unseen words only. The only notable deviation from the general

	TB-SBX	TB-UD	Euk	Euk 4-fold
KB-Bert	**97.71** (0.2)	**97.28** (0.1)	**96.64** (1.1)	**97.14** (0.4)
Flair	97.31 (0.2)	96.79 (0.1)	95.88 (1.6)	96.63 (0.5)
Stanza	96.18 (0.3)	95.79 (0.1)	94.64 (1.7)	95.39 (0.8)
Marmot	95.62 (0.4)	94.94 (0.2)	93.75 (2.1)	94.72 (1.0)
Hunpos	93.58 (0.5)	92.85 (0.2)	91.31 (2.5)	92.33 (1.5)

Table 3: 5-fold macroaveraged accuracy for POS+MSD for all three corpora and all five taggers (standard deviation in parentheses). The final column shows a 4-fold macro average for Eukalyptus, excluding the blog part for testing.

TB-SBX	TB-UD	Euk
97.94 (0.05)	97.36 (0.11)	97.42 (0.04)

Table 4: 5-fold macroaveraged accuracy for POS+MSD for all three corpora using KB-Bert, where the data has been divided over the folds randomly (SD in parentheses).

	TB-SBX	TB-UD	Euk
KB-Bert	**72.69** (4.5)	**68.83** (3.4)	**59.86** (5.2)
Flair	68.98 (4.9)	64.47 (2.7)	54.15 (5.8)
Stanza	60.10 (5.0)	57.55 (2.8)	46.27 (5.1)
Marmot	55.31 (4.6)	51.11 (2.6)	40.84 (5.2)
Hunpos	45.47 (4.4)	39.99 (2.1)	31.86 (5.4)

Table 5: 5-fold macroaveraged sentence-level accuracy for POS+MSD for all three corpora and all five taggers (SD in parentheses).

	TB-SBX	TB-UD	Euk
train	3377 (319)	3246 (257)	4368 (723)
train-dev	3076 (270)	2948 (242)	4065 (717)

Table 6: Average numbers of unseen words for the 5-fold test data sets (SD in parentheses). The train-dev data was used for training Hunpos and Marmot, while the train data only was used for KB-Bert, Flair, and Stanza.

	TB-SBX	TB-UD	Euk
KB-Bert	**93.31** (0.4)	**92.90** (0.4)	**91.21** (3.2)
Flair	92.65 (0.6)	92.17 (0.4)	89.36 (3.8)
Stanza	88.65 (1.0)	88.49 (0.6)	85.33 (4.5)
Marmot	87.78 (0.9)	86.96 (0.7)	82.68 (5.8)
Hunpos	82.68 (3.5)	82.68 (3.2)	82.68 (12.6)

Table 7: 5-fold macroaveraged results for POS+MSD for previously unseen wordforms for all three corpora and all five taggers (SD in parentheses).

results is that Hunpos does equally well on unseen words for all three corpora. Given that Eukalyptus exhibits a large variation of unseen words, we examine the results per split. The results for the Blog fold are the worst (about 10 points lower POS+MSD-tagging accuracy on OOV tokens than the rest of the folds), while the number of OOV tokens in this fold is relatively low. This indicates that the unseen words in the blog data are difficult to tag given the context.

4.4 Difficult categories

If we look at the top-3 and bottom-3 POS tags, ranked by F1-score, for each fold and each tagger, we see that for Eukalyptus the worst tags are foreign words, interjections and proper nouns. Adverbs and adjectives appear among the bottom 3 once each (over all testfolds and all taggers). For Talbanken-SBX and Talbanken-UD the bottom is not as clear. The most frequent in the bottom 3 for Talbanken-SBX are foreign word, verb particle and interjection, while proper nouns, possessive wh-pronouns and wh-determiners appear a few times. Participles and ordinals appear only once. For Talbanken-UD symbols, subordinating conjunctions, interjections and proper nouns appear in the bottom 3 most frequently, while adverbs appear only twice.

Overall, this shows that interjections, foreign words, and proper nouns are difficult to predict correctly. This may not be surprising, since these categories generally apply to words with a high type count and there are no visible morphological cues. Foreign words additionally have a wide range of syntactic functions. Note that UD has a feature (MSD-tag) for foreign words, but not a POS-tag.

Another reason for these categories being difficult, at least in part, is that they are infrequent. Let us therefore explore categories with higher frequencies. Considering that there are generally around 20,000 tokens in the test sets, we can look at categories with more than 200 instances in the test data (ignoring categories with less than 1% of the test tokens each).

We see that for Eukalyptus, proper nouns, adjectives and adverbs are generally difficult, with foreign words, conjunctions and nouns also appearing in the bottom 3 at times. Hunpos seems to have more problems with nouns, however. Marmot has less difficulties with nouns, instead finding numerals slightly difficult. For Talbanken-SBX, participles are difficult, as well as proper nouns, adjectives and adverbs. Bert seems to also have problems with cardinals, but less with adverbs, while Marmot has less trouble with adjectives. For Talbanken-UD, the most difficult categories are proper nouns and subjunctions. Adverbs are also difficult for most taggers, although less so for Hunpos. Auxiliaries are a bit more difficult for Marmot and Hunpos, while numerals are bit more difficult for Bert, Flair and Stanza. Altogether, these differences can be exploited, for example in an ensemble approach (Section 4.6).

Looking at POS+MSD confusion matrices, we can see that one of the most frequent confusions (especially for both Talbankens) is that of singular and plural neuter indefinite nouns (in both directions). Indefinite singular and plural forms for Swedish neuter nouns ending in a consonant are syncretic (*barn* 'child/children', *hus* 'house/houses'). The problem is exacerbated by the fact that at least in Talbanken-SBX, there are many contexts where the number of the noun cannot actually be inferred (both interpretations are possible). Such nouns, however, are not annotated as underspecified for number, but as either singular or plural, often inconsistently, which makes learning difficult. One example is shown in the example below. *Undantag* is tagged as plural according to the gold data, and as singular by KB-Bert, and both interpetations are possible.

(1) *Undantag: periodiskt understöd eller*
Exception(s): periodic support or
därmed jämförlig periodisk inkomst
comparable periodic income

In Talbanken-UD, a frequent error concerns confusing verbs and auxiliaries. It seems to be that the distinction between these two categories is not entirely consistently annotated in Talbanken-UD. In the following shortened examples, the gold data has different annotations for the verb *vara* 'be', although there is no clear difference between the two.

(2) *Frågan är [AUX] om*
The question is if
man med den konservativa grundsynen kan [...]
one with the conservative basic view can

(3) *Frågan är [VB] om*
The question is if
synen på äktenskapet kan [...]
the view of marriage can [...]

An issue particular to Eukalyptus is confusing symbols and punctuation. They are considered the same POS category, but two different MSD tags. This is not very surprising and seems to emerge from the amount of smileys in the blog fold. The result is a frequent mistagging of symbols as punctuation in the blog fold, and several cases of mistagging punctuation as symbols in the other folds, in particular in the novels. Many of the latter cases are quotation dashes, indicating a character's speech. This method of marking direct speech is uncommon in the other types of texts.

4.5 What makes a tag difficult: quantitative analysis

We also perform a systematic statistical analysis of the factors which can potentially affect tagger performance. More specifically, we attempt to identify which properties make a tag difficult.

For every corpus, we concatenate all five test sets (i.e. microaverage across folds), and measure the following for every POS+MSD tag:

- the accuracy of every tagger on this tag;
- the frequency. The prediction is that frequent tags are easier to identify;
- type-token ratio (TTR) of tokens that have this tag. The prediction is that high TTR will make the tag more difficult to identify, cf. Section 4.4. TTR is strongly dependent on the sample size (less frequent tags are more likely to have higher TTR), but we judge that in this case, no correction is necessary;
- average "difficulty" of tokens that have this tag. This is done in two steps. First, we go through all tokens in the dataset, calculate the probability distribution of tags for every token and then the Shannon entropy of this

Predictor	Average (%)	SD	Significance
Frequency	0.003	0.0006	10/15
TTR	-85.2	6.4	15/15
Tag-by-token entropy	-27.0	7.4	15/15
Tag-by-ending entropy	6.8	3.1	10/15

Table 8: Summary of the regression models: average slope values and SD across all 15 models. Significance shows in how many of the models the predictor is significant at 0.05 level.

distribution. The entropy shows for every token how difficult it is to guess its tag and thus serves as a measure of "token difficulty". At the second step, when analyzing a particular tag, we weigh the associated entropy by the relative frequency for every token that has this tag. We then sum the weighted values. The result (average conditional entropy) is meant to gauge how difficult on average the tokens that have the particular tag are;

- average "difficulty" of token endings (average entropy of tag conditioned on token ending). The procedure is exactly the same as for tokens, but instead of the whole token we are using its ending, which is typically the main grammatical marker in Swedish. For instance, *-er* can mark a present-tense verb or an indefinite plural noun. We are using the last two characters of the token as the ending (or the whole token if it's shorter than two characters).

We fit a linear regression model with accuracy as the dependent variable (measured as percentage, i.e. on the 0–100 scale) and the four predictors described above as independent variables. We fit a separate model for every tagger and every corpus, i.e. 15 models in total. For all corpora, the collinearity of the predictors is very mild (the condition number varies from 8.2 to 9.5) and thus acceptable (Baayen, 2008, p. 181–182).

We summarize the results of the 15 models in Table 8. The results are very similar across corpora and folds for TTR and tag-by-token entropy, less so for frequency and tag-by-ending entropy. All models have high goodness-of-fit: the average multiple R^2 is 0.65, SD is 0.05.

In general, the first three predictions are borne out. On average, the increase in frequency by 1 token is expected to result in the increase in the tag accuracy by 0.003%. Frequency ranges from 1 to 11,000, which means that theoretically, the largest expected increase can be 33%.

The increase in tag-by-token entropy by 1 (note that this is a very large increase: entropy varies from 0 to 1.86 in our sample) is expected to decrease accuracy by 27%. The increase in TTR by 1 is expected to decrease accuracy by 85.2% (note that TTR cannot actually be larger than 1). TTR that is close to 0 is typical for tags that are assigned to a very small closed class of frequent tokens (e.g. punctuation marks). TTR of 1, on the contrary, can be achieved by tags that occur with (a few) very infrequent tokens (this is often a result of misannotation, or some very infrequent form or usage).

Surprisingly, the average conditional entropy of the tag given the ending goes directly against the prediction, yielding a positive effect (though small and not always significant). We cannot explain this effect. Our best guess is that high tag-by-ending entropy is correlated with some other properties that facilitate accurate tagging.

4.6 Ensemble

We tested whether combining the output of the five taggers may yield improved performance. In theory, it should be possible, since the proportion of cases where *at least one of the taggers* outputs a correct tag is higher than the accuracy of any individual tagger (see Table 9, row "Ceiling").

We tried simple voting and a naive Bayes classifier (as implemented in the NBayes Ruby gem [12]). In both methods, the taggers are ordered by performance in descending order. In simple voting, each tagger gets one vote. In case of a tie, the vote that has come first wins. The naive Bayes classifier has to be trained. For that, we split the test set in each fold of each corpus into a training set (75%) and a test set (25%). What the classifier learns is how to match the input string (the token and the tags proposed by each tagger) with the label (which tagger makes the correct guess). If several taggers make a correct guess, the first one of those is chosen. If

[12] `https://github.com/oasic/nbayes`

Method	TB-SBX	TB-UD	Euk
Ceiling	99.16 (0.1)	98.78 (0.4)	98.26 (1.5)
KB-Bert	97.65 (0.3)	97.35 (0.6)	96.72 (1.6)
Voting	97.50 (0.1)	97.12 (1.0)	96.38 (2.2)
Bayes	97.65 (0.2)	**97.41** (0.5)	96.75 (1.6)
Voting-fast	96.96 (0.3)	96.69 (0.7)	95.91 (1.8)
Bayes-fast	**97.67** (0.3)	97.37 (0.6)	**96.76** (1.7)

Table 9: Results of ensemble methods with comparison to the potential ceiling (at least one of the taggers guessed right) and the best single tagger (macroaveraged accuracy across all folds, SD in parentheses).

no taggers make a correct guess, KB-Bert is chosen by default. Changing this method (e.g. using only the tags as the input string) leads to slightly worse performance. Both voting and the classifier are then tested on the test set. Since Stanza and Flair are slow at training time, we also try a combination of the "fast" taggers: KB-Bert, Marmot and Hunpos.

The results are summarized in Table 9. Simple voting always performs worse than the best single tagger, but naive Bayes performs slightly better. For Talbanken-SBX and Eukalyptus, the best performance is achieved when the classifier is trained on the output of fast taggers only, while for Talbanken-UD the full training set yields better results. All differences are, however, very small. The difference between KB-Bert and Bayes is not significant ($t(14) = -1.1$, $p = 0.28$, d = -0.03), nor is the one between KB-Bert and Bayes-fast ($t(14) = -1.6$, $p = 0.12$, d = -0.03), no correction for multiple comparisons.

A possible avenue for future research would be to use other recently developed ensemble methods, as for instance Bohnet et al. (2018); Stoeckel et al. (2020).

5 Conclusions

We applied five taggers to three important Swedish corpora. The corpora are of comparable size and have different tagsets. Two of them consist of virtually the same texts, but are not entirely parallel.

We show that the three neural taggers outperform the two pre-neural (HMM and CRF) ones when it comes to tagging quality, but are significantly slower. KB-Bert, however, while always yielding the highest accuracy, is also the fastest of the neural taggers, and its speed on GPU is comparable with that of the pre-neural taggers.

Token-level accuracy of KB-Bert (97.2 on average across corpora) is very high, and is decent also for OOV tokens (92.5). If we apply sentence-level accuracy, a less forgiving measure (Manning, 2011), we can see that there is actually much room for improvement (67.1).

The success of the taggers depends to a large extent on the additional data (embeddings, morphological dictionaries) that they receive as input, of which token embeddings (a.k.a. contextualized or dynamic) seem to be the most powerful ones. It is reasonable to assume that it is also important on which corpus the embeddings were trained. The size of this corpora is comparable for all neural taggers, but KB-Bert's is likely to be the most balanced one.

The results vary across corpora/tagsets. If we use consecutive splits, TalbankenSBX always has the highest annotation accuracy and Eukalyptus the lowest one. The reason for that is that the two Talbankens are more homogeneous (contain only professional prose texts), while Eukalyptus contains texts from five different domains, one of which (blogs) is notoriously difficult. The reason for TalbankenSBX yielding better results than TalbankenUD is probably the less fine-grained tagset, but possibly also more consistent annotation. If, however, we use random splits, the accuracy for Eukalyptus goes up, surpassing the one for TalbankenUD.

Manual error analysis suggests that a high type count, absence of morphological cues, a wide range of syntactic functions, and low frequency make tags more difficult. Inconsistent annotation (which is very difficult to avoid in borderline cases) also seems to play an important role. We also perform a statistical analysis of the factors that can potentially affect how difficult the POS+MSD tags are. The regression model shows that type-token ratio within tag and average "difficulty" of tokens within tag (measured as entropy of guessing the tag given the token) have consistently significant and very strong negative effects on the accuracy. Tag frequency has a positive (though not always significant) effect. Surprisingly, so does the average "difficulty" of token endings within tag (though the effect is small and not always significant). The results of the statistical analysis partly support the predictions done on the basis of the manual one. In general, this is a promising research avenue which deserves more systematic attention.

Finally, we test whether the tagger outputs can be combined using ensemble methods, since in theory, there clearly is a potential for that. In practice, it turns out that using a naive Bayes classifier it is possible to achieve a very small improvement over the best-performing tagger, but the difference is not statistically significant.

The data and scripts that are necessary to reproduce the regression analysis and the ensemble methods are available as supplementary materials[13].

Acknowledgments

This work has been funded by Nationella Språkbanken – jointly funded by its 10 partner institutions and the Swedish Research Council (2018–2024; dnr 2017-00626). We would like to thank Gerlof Bouma, Simon Hengchen and Peter Ljunglöf for valuable comments on earlier versions of this paper.

References

Yvonne Adesam and Gerlof Bouma. 2019. The Koala part-of-speech tagset. *Northern European Journal of Language Technology*, 6:5–41.

Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania. Edited by Beáta Megyesi*, pages 1–9.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Harald Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In Swedish Language Technology Conference (SLTC). Umeå University.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language resources and evaluation*, 47(4):1191–1211.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp – the corpus infrastructure of språkbanken. In *Proceedings of LREC 2012. Istanbul: ELRA*, page 474–478.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project - description and guidelines. Technical Report 33, Department of Linguistics, Umeå University.

Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of Swedish word vectors. In *Sixth Swedish Language Technology Conference (SLTC), Umeå 17-18 nov 2016.*

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, page 209–212, USA. Association for Computational Linguistics.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer.

[13] https://github.com/AleksandrsBerdicevskis/Swetagging2021

Beáta Megyesi. 2009. The open source tagger HunPoS for Swedish. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 239–241, Odense, Denmark. Northern European Association for Language Technology (NEALT).

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Joakim Nivre. 2014. Universal Dependencies for Swedish. In Swedish Language Technology Conference (SLTC). Uppsala University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA).

Joakim Nivre and Beata Megyesi. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories*, pages 97–102.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1392–1395. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Manuel Stoeckel, Alexander Henlein, Wahed Hemati, and Alexander Mehler. 2020. Voting for POS tagging of Latin texts: Using the flair of FLAIR to better ensemble classifiers by example of Latin. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 130–135, Marseille, France. European Language Resources Association (ELRA).

Anders Søgaard, Sebastian Ebert, Joost Bastings, and Katja Filippova. 2020. We need to talk about random splits.

Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.