A Challenging Data Set for Evaluating Part-of-speech Taggers

Citation for the original published paper (version of record):
Wahde, M., Suvanto, M., Della Vedova, M. (2024). A Challenging Data Set for Evaluating
Part-of-speech Taggers. International Conference on Agents and Artificial Intelligence, 2: 79-86.
http://dx.doi.org/10.5220/0012307200003636

(article starts on next page)

A Challenging Data Set for Evaluating Part-of-Speech Taggers

Mattias Wahde [a], Minerva Suvanto [b] and Marco L. Della Vedova [c]
Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

Keywords: Part-of-Speech Tagging, Natural Language Processing, Sequence Labeling.

Abstract: We introduce a novel, challenging test set for part-of-speech (POS) tagging, consisting of sentences in which only one word is POS-tagged. First derived from Wiktionary, and then manually curated, it is intended as an out-of-sample test set for POS taggers trained over larger data sets. Sentences were selected such that at least one of four standard benchmark taggers would incorrectly tag the word under consideration for a given sentence, thus identifying challenging instances of POS tagging. Somewhat surprisingly, we find that the benchmark taggers often fail on rather straightforward instances of POS tagging, and we analyze these failures in some detail. We also compute the performance of a state-of-the-art DNN-based POS tagger over our set, obtaining an accuracy of around 0.87 for this out-of-sample test, far below its reported performance in the literature. Also for this tagger, we find instances of failure even in rather simple cases.

1 INTRODUCTION

Part-of-speech (POS) tagging is an important preprocessing step in many natural language processing (NLP) tasks (Chiche and Yitagesu, 2022). In POS tagging, the aim is to assign class labels (POS tags) to the words in a sentence, determining, for each word, whether it is a noun, a verb, an adjective, and so on. For many words, this is simple, as they are always associated with a single POS tag. For example, the word *organization* is always a (common) noun. On the other hand, there are many words for which several different POS tags are possible, so that the correct tag in a given situation must be inferred from context, i.e., using information about surrounding words. For example, the word *present* can be either a noun (meaning a *gift*), a verb (as in *present a paper at a conference*), or an adjective (as in *being present at a meeting*). The aim of a POS tagger is to resolve such ambiguities and thus to assign the correct POS tag to each word in a sentence.

POS tagging has been extensively studied in NLP, resulting in a set of high-performance taggers, such as Brill (Brill, 1992), Hunpos (Halácsy et al., 2007), Perceptron (Bird et al., 2009), Stanford (Toutanova et al., 2003), and, more recently, a variety of taggers using deep neural networks (DNNs), see, e.g., (Akbik et al., 2018). Achieving decent POS tagging performance is not a very difficult task. In fact, when applied to a diverse set of sentences, a tagger that simply selects the most common tag for any given word, typically obtains an accuracy of at least $0.85-0.90$.

More sophisticated taggers, as exemplified above, generally obtain even better results, with reported accuracy in the range $0.92-0.97$. Recent DNN-based taggers typically exhibit high accuracy, but other taggers, some of which are listed above, are not far behind. Indeed, the reported accuracy of the Stanford tagger (over its test set) is 0.972 (Toutanova et al., 2003), whereas the performance of a state-of-the-art DNN-based tagger (Akbik et al., 2018) is 0.978 over the same set. Here, one should also bear in mind that the ground truth labels involve some error (or, at least ambiguity), where different human evaluators may assign different POS tags, typically affecting around 3% of the words[1]. Thus, one can hardly demand an accuracy of better than around 0.97 of any automated tagger. For this reason, given the accuracies mentioned above, one may perhaps view POS tagging as a *solved* problem and thus simply apply any of the standard POS taggers included in commonly used software libraries.

However, we argue that such a conclusion may be premature. First of all, most (English) POS taggers have been trained on one of two specific data sets,

[a] https://orcid.org/0000-0001-6679-637X
[b] https://orcid.org/0009-0003-1751-151X
[c] https://orcid.org/0000-0002-4703-7500

[1] The topic of POS ambiguity is further discussed in Section 2 below.

Wahde, M., Suvanto, M. and Vedova, M.
A Challenging Data Set for Evaluating Part-of-Speech Taggers.
DOI: 10.5220/0012307200003636
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 2*, pages 79-86
ISBN: 978-989-758-680-4; ISSN: 2184-433X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

namely the Brown corpus (Francis and Kucera, 1979) and the Penn Treebank (Marcus et al., 1999). Even though those are excellent data sets, there is still a risk of overstating the out-of-sample performance: If a sufficient number of different POS taggers are generated, some will naturally perform better than others on the parts of the data set used for testing, but those results are not necessarily replicated on completely new data.

Second, the sentences in the two data sets are now around 30 years old (Penn Treebank) and more than 50 years old (Brown). Over such a long time span, any human language will undergo changes that, in turn, may lead to reduced POS accuracy. Third, even though the average performance of POS taggers is very good, there are many cases (as will be discussed in Section 6 below) where existing POS taggers fail, not only in complicated cases involving tags that are hard to assign even for a human, but also, in fact, in simple cases.

For these reasons, we here introduce a new *test* set for POS tagging, which has been semi-automatically generated using Wiktionary, followed by thorough manual inspection, as discussed below. This data set specifically targets word usages for which a set of representative POS taggers struggle. It should only be seen as a *test* set since, as a result of its method of construction, we only tag *one* word per sentence. Nevertheless, the new data set offers an opportunity to test a POS tagger (trained on any other data set) in an out-of-sample fashion.

The paper is structured as follows: First, in Section 2 we begin with some general observations related to POS tagging. Next, in Section 3 we introduce and describe the new POS tagging test set. Then, in Section 4, we list and describe the taggers used here. In Section 5 we present the results obtained for the various taggers, both over standard benchmark data sets and our data set. Then, in Section 6, we analyze different instances of misclassification by the various taggers. We also briefly present a rule-based approach (in development) that aims to improve tagger performance by focusing on those words or phrases where existing taggers struggle. Conclusions are given in Section 7.

2 POS TAGGING: OBSERVATIONS

As mentioned above, POS tagging is the problem of assigning a class label (POS tag) to each word in a sentence. In other words, the process maps an ordered sequence of words $\{w_1, w_2, \ldots, w_n\}$ to a sequence of tuples (w_i, p_i), where p_i are the POS tags. Note that there are related procedures, such as *chunking* (Wu et al., 2023; Ramshaw and Marcus, 1999) that seeks to divide a text into units that may contain more than one word, for example noun phrases, verb phrases, compound nouns, and so on. In this paper, however, we will consider only POS tagging, as defined above.

Given a *tag set*, i.e., the set of possible labels, POS tagging is straightforward in many cases. Consider, as an example, the simple sentence *she slowly opened the heavy door*. In this sentence, *she* is a personal pronoun (denoted PP in the Penn Treebank tag set), *slowly* is an adverb (RB) *opened* is a verb in past tense (VBD), *the* is a determiner (DT), *heavy* is an adjective (JJ), and *door* is a noun in singular form (NN), resulting in the following sequence (*she*, PP), (*slowly*, RB), (*opened*, VBD), (*the*, DT), (*heavy*, JJ), (*door*, NN).

However, in many other cases, the process is less straightforward. First of all, there may be errors in the data sets used for training POS taggers. As mentioned above, it is typically stated that the accuracy of ground truth tags is around 97% but, as already noted by (Manning, 2011), this number may be an overestimate. To some degree, the ground truth POS tags could perhaps be improved via a time-consuming process of manual correction, though. Manning proceeds to break down POS tagging errors (for the Stanford tagger) into several categories, for example *lexicon gap* where a word appears in the training set, but never with the tag relevant in the test sentences, *difficult linguistics* where assigning a correct tag depends on long-range context unavailable to a tagger based on local features, and *underspecified*, where the tag is simply unclear or ambiguous.

There are indeed cases of POS ambiguity. Consider, for instance, the sentence *she was surprised when she came home*. In this case, the word *surprised* can be taken as a verb in passive form, describing an action or event (as in: *As she entered her house there was something that surprised her*), but it could *also* be an adjectival form, describing a state (as in: *something surprised her earlier in the day, and she then remained in a surprised state when coming home*). In such cases, one simply *cannot* assign a unique, correct tag without additional context.

Compound nouns are another example: While in some languages, e.g., Swedish, Finnish, German, and so on, compound nouns are normally closed, in English many compound nouns are *open*, such as *full moon*, *fire truck*, *waiting room*, *high school*, *hot dog*, *free trade* and so on. In the last example, *hot dog* is normally a compound noun, describing a type of food rather than the state of a dog. Yet, considered separately, *hot* is an adjective here so should it, in POS tagging, be marked as such (as indeed the Stanford tagger does) or should it be marked as a noun, noting that it is a part of a compound? One can make a similar observation for the case of *free trade*, where the Stanford tagger marks *free* as an adjective.

In the cases described above, it is not surprising that a POS tagger may assign a tag that differs from the (ambiguous) ground truth. However, as will be illustrated below, there are also many cases which are quite straightforward where even very good taggers make rather elementary and surprising errors, indicating that POS tagging is still far from solved.

Moreover, it possible that the performance of a given POS tagger will degrade with time, as a result of changes in language. Such *temporal drift* effects have been found in the related case of named-entity recognition (NER) (Liu and Ritter, 2022). With the rapid development of new technology, vocabulary changes in several different ways: Completely new words are introduced (for example the verb *to google*), whereas other words change their meaning, for example the verb *tweet* that, today, more commonly means posting something on (the now renamed) Twitter, rather than a bird making a sound. Another problem is that new POS tags may come to be used for a given word. For example, especially in words related to new technology, it is not uncommon to use as a verb a word that was previously almost exclusively used as a noun, for example the word *text*. This poses a problem especially for POS taggers whose training is data-driven: For those taggers it is sometimes difficult to tag correctly words that are used in a manner not seen in the training set. Next, we will introduce our data set, which highlights some of the problems listed above.

3 A NEW TEST SET FOR POS TAGGING

The purpose of the new POS test set[2] is that it should act as challenging test for existing POS taggers. The data set was generated as follows: First, an offline copy of Wiktionary was parsed to extract sample sentences for the many words contained in the dictionary. Now, Wiktionary is structured such that, for any given word, the different usages (as a verb, noun, adjective, and so on) are organized into separate sections, and are typically associated with one or several examples, making it possible to automatically extract a large number of sentences, for which *one* word has a specified POS tag. The structure of the (HTML) pages is not entirely consistent over the entire Wiktionary, somewhat complicating the automated extraction of such sentences. Nevertheless, our extraction process, which will not be described in detail here, resulted in roughly 67,000 sentences.

Table 1: Some basic statistics for the new POS data set. Here, *n* denotes the number of instances for the tag in question. Note that NUM, DET, PART, and X do not appear among the tagged words (one per sentence) in our data set. In total there were 2,227 sentences and, therefore, 2,227 POS tags.

POS tag	*n*	Fraction
VERB	977	0.4387
NOUN	578	0.2595
ADJ	524	0.2353
ADV	105	0.0471
ADP	16	0.0072
CONJ	16	0.0072
PRON	11	0.0049

Using only the Wiktionary data, there is no simple way to assign fine-grained POS tags, e.g., distinguishing between different verb forms, since the examples are all grouped in a broader category (e.g., *verb*, in that case). Thus, for our data set, we have chosen to use the universal POS tags (Petrov et al., 2012), namely NOUN (common noun), VERB, ADJ (adjective), ADV (adverb), ADP (adposition, meaning preposition or postposition), CONJ (conjunction), and PRON (pronoun). There are additional universal POS tags, namely NUM (number), PART (particle), X (unknown), DET (determiner), but they did not feature in our data, and neither did PUNCT (punctuation).

This set was then manually curated, removing any sentences involving a POS tag that was either deemed ambiguous (e.g., cases involving passive verbs that could equally well be taken as adjectival forms) or simply incorrectly assigned in Wiktionary. From the remaining sentences, a drastic reduction was carried out: Sentences were kept only if at least one of the four benchmark taggers (listed in Section 4.1 below) assigned an incorrect POS tag, resulting in a set with 2,227 sentences: Even though the four benchmark POS taggers were correct in most cases (roughly in accordance with their general performance estimates; see Table 3), here we are specifically interested in cases where those taggers *fail*. Some basic statistics for our data set are shown in Table 1, whereas some examples of sentences are shown in Table 2.

As noted above, for every sentence in our data set, only a single word is POS tagged. Clearly, as the benchmark taggers make use of surrounding words when assigning POS tags, our data set cannot be used for *training* a POS tagger, but it can be used for *testing* an existing tagger. The procedure is as follows:

[2] Available at https://doi.org/10.5281/zenodo.10299108

Table 2: Some examples of sentences from our data set. The POS-tagged word is shown underlined, in **bold**. The ground truth tag is given in the second column, and the tags assigned by the four benchmark taggers are given in the remaining columns. Note that many sentences in the data set are considerably longer; short sentences were chosen here simply to fit in the table. The DNN-based tagger (see Section 4.2) correctly tagged the first three sentences, but failed for the two last.

Sentence	Ground truth	Brill	Hunpos	Stanford	Perceptron
How does this **bear** on the question ?	VERB	NOUN	NOUN	VERB	NOUN
Seaweed **clung** to the anchor .	VERB	VERB	VERB	NOUN	NOUN
That doctor is nothing but a lousy **quack** !	NOUN	NOUN	PART	X	NOUN
When you **quiet** , we can start talking .	VERB	ADJ	ADJ	VERB	VERB
He bought a **used** car .	ADJ	VERB	ADJ	VERB	ADJ

For every sentence, the tagger is applied to the entire sentence, assigning tags to each word. Next, the assigned tag for the single POS-tagged word (for a given sentence) is mapped to the universal tag set, and can then be compared to the ground truth tag, from our data set. Thus, even though the set is rather small, it is large enough to provide interesting insight into the reasons for failures of various taggers.

4 POS TAGGERS

In Section 4.1 we list and describe four commonly used POS taggers that, as mentioned above, were used when defining our POS data set: For every sentence in our data set, at least one of the four taggers failed to assign a correct POS tag to the word under consideration.

Using the procedure just described, one obtains a data set for which the absolute performance (of the four benchmark taggers) does not convey much information, since the sentences were indeed chosen deliberately to make those taggers fail. Thus, in addition, in Section 4.2 we consider also a recent, state-of-the-art tagger (Akbik et al., 2018) based on Bi-LSTMs.

4.1 Benchmark taggers

The benchmark taggers that we have used for defining our data set are the Brill, Hunpos, Stanford, and Perceptron taggers. We chose these four high-quality, frequently used POS taggers for comparison because they represent a variety of different approaches to POS tagging. In all cases except Brill, a standard pre-trained version is available and free to use. We also included the Brill tagger despite the absence of a pre-trained version, since it is an influential and much-used POS tagger, and because it is rule-based, while the others are primarily statistical. In what follows we provide a brief description of each of these taggers.

4.1.1 Brill Tagger

The Brill tagger (Brill, 1992), is one of the pioneering POS taggers in the field of computational linguistics. It is based on transformation-based learning, a rule-based approach that iteratively refines tag assignments by applying a set of transformation rules to the initial tag sequence. These rules are learned from annotated training data and aim to correct tagging errors progressively. The Brill tagger achieved state-of-the-art accuracy during its time and served as a foundation for subsequent tagger development. Since there is no standard pre-trained version of the tagger, we trained the model similarly to the original version over a random sample (50%) of the Brown corpus. In particular, we used a unigram tagger with a simple regular-expression-based backoff as initial tagger and the original 24 templates defined in (Brill, 1992).

4.1.2 Hunpos Tagger

The Hunpos tagger is a statistical POS tagging tool that employs hidden Markov models (HMMs) to predict the most likely tag sequence for a given input sentence (Halácsy et al., 2007). It is an open source reimplementation of the *Trigrams'n'Tags* (TnT) tagger (Brants, 2000). The tagger leverages both word-level and contextual information, taking into account the probabilities of transitions between POS tags and the emission probabilities of words given their tags. We considered the current pre-trained version (v.1.0-en_wsj)[3], which has been trained on the Wall Street Journal (WSJ) section of the Penn Treebank dataset, with the usual division into training, validation, and test sets (Collins, 2002).

4.1.3 Stanford Tagger

The Stanford POS tagger, developed by the Stanford NLP Group, is a widely used POS tagging tool that combines both rule-based and probabilistic approaches (Toutanova et al., 2003). It employs a maxi-

[3] Available at https://code.google.com/archive/p/hunpos/downloads (v.1.0-en_wsj).

mum entropy model to assign tags to words based on features such as word identity, word shape, and surrounding words. We used the current pre-trained version (v4.2.0)[4]. Like the Hunpos tagger, the Stanford tagger was trained on the Penn Treebank dataset.

4.1.4 Perceptron Tagger

The Greedy Averaged Perceptron tagger written by Honnibal is currently the standard tagger in the Python NLTK (Natural Language Toolkit) library[5] (Bird et al., 2009) and, as such, is frequently used as a default option in POS tagging. The perceptron algorithm is a linear classification algorithm that iteratively updates weights to train a discriminative model for assigning tags to words. We considered the current pre-trained version (v.3.8.1) present in the library, which, like the previous two taggers, has been trained on the Penn Treebank corpus.

4.2 DNN-based Tagger

Given that our data set was defined by deliberately choosing sentences where at least one of the four benchmark taggers failed to tag the selected word correctly, the performance obtained for those four taggers will, of course, be artificially reduced. Thus, in order to make a true out-of-sample test of POS tagger performance over our set, we also considered a tagger based on deep neural networks (DNNs), more precisely the Bi-LSTM tagger presented by (Akbik et al., 2018), which has a reported accuracy of 0.978 over the Penn Treebank test set, using the data division described in (Collins, 2002). There are many other DNN-based taggers, see, for example, (Yasunaga et al., 2017; Ling et al., 2015), but their reported performance is not much different from that reported in (Akbik et al., 2018). Thus, for our purposes, it is sufficient to consider only this tagger.

5 RESULTS

In this section we present the POS tagging performance of the four benchmark taggers and the DNN-based tagger, over different sets, namely a group of randomly selected subsets of Brown and Penn Treebank, as well as our set.

Table 3: The performance (measured in terms of accuracy) of the four benchmark taggers over subsets of two standard POS data sets, mapped to the universal tag set. The ranges shown are the 95% confidence intervals, assuming a Gaussian distribution.

Tagger	Data Set	Accuracy
Brill	Brown	0.966 ± 0.001
Brill	Penn Treebank	0.903 ± 0.001
Hunpos	Brown	0.919 ± 0.001
Hunpos	Penn Treebank	0.987 ± 0.000
Stanford	Brown	0.940 ± 0.002
Stanford	Penn Treebank	0.967 ± 0.000
Perceptron	Brown	0.917 ± 0.001
Perceptron	Penn Treebank	0.974 ± 0.000

Table 4: The performance of the four benchmark taggers over the new POS data set, measured in terms of accuracy.

Tagger	Accuracy
Brill	0.473
Hunpos	0.482
Stanford	0.673
Perceptron	0.561

5.1 Benchmark Tagger Performance

The benchmark taggers were typically trained on one of the two standard POS tagging data sets mentioned in Section 1, using holdout validation with a training set, a validation (development) set, and a final test set. The reported results generally refer to the performance of each tagger over the test set. Since the taggers were trained on different data sets (with varying training-validation-test splits) and using different tag sets, it is difficult to make an exact, direct comparison. Thus, here, we chose to generate five subsets, each with 2,000 randomly selected sentences, from either the Brown data set or the Penn Treebank data set, thus giving a total of 10 subsets. We then computed the performance of our four benchmark taggers over those subsets, using their original tag sets during tagging, and then mapping the results to the universal POS tags (Petrov et al., 2012). The results are shown in Table 3. As can be seen, the performance is generally quite good, with accuracies in the range 0.90 - 0.99, and in all cases with very small variation over the subsets.

Looking at our POS test set, where the accuracy is measured based on the ability of the taggers to tag the selected word (see also Table 2), the situation is quite different, as can be seen in Table 4. Now, it is not surprising that the performance is much lower than that shown in Table 3, since our data set was defined by deliberately selecting sentences where at least one of the benchmark POS taggers fails. Thus, while the absolute performance values in Table 4 are

[4] Available at https://nlp.stanford.edu/software/tagger.shtml, *english-bidirectional-distsim* model (v.4.2.0).

[5] See https://www.nltk.org (version 3.8.1).

Table 5: The performance of the DNN-based tagger over the same subsets (first and second row) used in Table 3, and over our POS data set (last row). In all cases, the results refer to the universal tag set.

Data set	Accuracy
Brown	0.949 ± 0.001
Treebank	0.972 ± 0.000
Our data set	0.868

not so relevant, we note the rather large *relative* performance differences of these four taggers, which can be compared with their about-equal performance in Table 3. Perhaps more importantly, and as analyzed in Section 6 below, they fail in many cases where the POS tag assignment is, in fact, quite straightforward.

5.2 DNN-based Tagger Performance

Turning now to the DNN-based tagger (Akbik et al., 2018), Table 5 shows the results obtained over the same 10 subsets used for the four benchmark taggers (Table 3), using the Penn tag set and mapping the final results to the universal POS tags (Petrov et al., 2012). As can be seen in the table, the performance over the 10 subsets fell roughly in the range $0.95-0.97$. Thus, in most cases (though not all), the DNN-based tagger does a bit better than the four benchmark taggers (see Table 3). By contrast, for our data set, the accuracy was 0.868, indicating that our data set is indeed challenging for the tagger.

6 DISCUSSION

The main aim of this paper has been to generate a novel challenging data set for POS tagging. Over such a data set, one would expect to find worse performance (lower accuracy) of even a high-quality DNN-based tagger, such as the one introduced by (Akbik et al., 2018). This is indeed what we find: Over our set, this tagger achieves an accuracy (using the universal POS tag set) of 0.868, quite a bit below its accuracy of roughly $0.95-0.97$ over the older, commonly used data sets, such as Brown and Penn Treebank.

One can also measure the performance of the very simplest POS tagger, namely the unigram tagger that, for every word, simply assigns the most frequent POS tag for the word in question. Using the vocabulary from the (entire) Brown data set to define the unigram tagger, it achieves an accuracy of only around 0.31 over our set, a number that can be compared with accuracies of 0.96 and 0.85 obtained when applying the same tagger over the Brown set and (a subset of) the Penn Treebank set, respectively. This dramatic drop in performance is partly due to the fact that the vocabulary for the unigram tagger was sourced from the Brown set, a set developed more than 50 years ago that contains no instances of many of the words in *our* data set. However, even if all such words are disregarded in the performance computation, the accuracy of the unigram tagger is only around 0.40 over our set, again illustrating that our data set is indeed challenging.

Besides these overall measures, it is interesting to make a more detailed analysis regarding failures of the various taggers over our data set, a topic that we will consider next. In that analysis, in addition to the DNN-based tagger, it is interesting to consider also the four benchmark taggers (especially Stanford) whose reported performance over their test sets is not far behind that of the DNN-based tagger (see Section 1), even though their performance over *our* set is of course lower, bearing in mind how our set was generated.

6.1 Analysis of Tagger Failure

The first thing to note is that, somewhat surprisingly, the taggers occasionally fail even in cases where the tag assignment is rather straightforward, as evidenced by the entries in Table 2. This applies also to the DNN tagger: While it correctly tagged the first three examples in that table, it failed on the last two.

Proceeding with a more detailed analysis, we note that some errors can occur due to similar morphological features shared between tags. For example, misclassifications between nouns and verbs happen in words ending in the suffix *-s* (*faces*, *casts*, *bears*) and *-ing* (*shipping*, *googling*, *ironing*). Similarly, the *-y* suffix is shared between nouns and adjectives (*accessory*, *cheesy*, *tidy*), and between adjectives and verbs we see the suffix *-ing* (*annoying*, *darling*) and *-ed* (*cursed*, *dated*, *requested*). The number of misclassified words with a given suffix is rather balanced between these tag pairs, with the exception of verbs and nouns, where the *-s* suffixed words are more often misclassified as nouns rather than vice versa.

Besides inspecting the spelling of words, taggers typically use surrounding words and their predicted tags (when available) to resolve ambiguities. The spelling of surrounding context words can also be similar between two classes, making it difficult for the taggers to choose the correct tag based on this information. For verbs and nouns, the preceding word *to* appears to be problematic. Here, the four benchmark taggers often fail to make the distinction between using *to* as a preposition combined with a noun, e.g., *I go to* ***church*** *every day*, and the to-infinitive, e.g., *It* *is time to* ***board*** *the aircraft*. In these examples, it is clear that *church* is a noun and *board* is a verb, yet the taggers struggle to distinguish between the two tags in these types of contexts. The DNN-based tagger performs better in these cases, but is not completely error-free either. For example, it tags the verb *bus* in the sentence *He was hired to* ***bus*** *tables* . . . as a noun.

Analyzing tags of surrounding words, we see that for adjectives mistagged as nouns the target word is often preceded by a determiner and followed by a noun. This sequence of tags (DET NOUN NOUN) is of course valid and it appears in the training sets, for example, for compound nouns (*a campaign coordinator*), two-part named entities (*the Ivory Coast*) and possessives (*her mother's assistant*). However, in the following tagged examples, taken from our data set, it is obvious that the mistagged word (whose tag is shown in red) does not belong to any of those categories. For example, in the sentence (*That*, DET), (*was*, VERB), (*a*, DET), (*classy*, NOUN), (*response*, NOUN), the word *classy* is clearly an adjective, even though some of the four benchmark taggers incorrectly tag it as a noun. Another example is the sentence (*He*, PRON), (*walked*, VERB), (*down*, ADP), (*the*, DET), (*lit*, NOUN), (*corridor*, NOUN), where *lit* is incorrectly tagged as a noun by the DNN tagger. It is not surprising that simple taggers that do not consider labels of next tokens (unidirectional taggers) would fail in such cases, but with the taggers evaluated here, it should not be a problem. We also see cases where some of the taggers do tag examples of this kind correctly, indicating that there are surprising inconsistencies. For example, if we replace the word *lit* in the previous sentence with *dark*, the DNN tagger classifies the word correctly as an adjective.

Incorrect labels might also be assigned when other tokens in the sequence have been misclassified first. In the following sentence, the Perceptron tags the word *neat* as a noun, and upon manual inspection, it seems to be because it follows the word *whisky*, which has been resolved as an adjective: (*I*, PRON), (*like*, VERB), (*my*, PRON), (*whisky*, ADJ), (*neat*, NOUN). It is clear that *whisky* is a noun in this context. We also note that the inspected token *neat* is correctly tagged by the DNN tagger, even though the previous token *whisky* is not: (*I*, PRON), (*like*, VERB), (*my*, PRON), (*whisky*, ADJ), (*neat*, ADJ). This type of sequence may be challenging due to the fact that, in English, adjectives are typically placed before the noun (attributive adjectives), but in this sentence the adjective *neat* appears after the noun (predicative and postpositive adjectives).

Many of the words that the taggers have failed to tag correctly include modern vocabulary related to, for example, computers, programming, finance and so on, indicating that there are some temporal drift effects causing misclassifications as well. The taggers are sometimes not able to generalize so as to handle either newly derived words (e.g., *hyperlink*, *botting*) or existing words adapted to a new meaning (e.g., the verb *text* as in *please text me when you are done.*).

Regarding smaller word classes included in our data set, only one pronoun is tagged correctly (by Hunpos and the DNN, in that case). None of the conjunctions are correctly tagged by any tagger. A possible explanation is that the data sets used to train the taggers do not include any examples where the words are used in this manner.

Whether the evaluated taggers learn general grammatical rules is questionable. We notice that around a third of the sentences where a verb is misclassified as a noun are tagged such that they contain no verb at all. A grammatically correct sentence typically has at least one verb, and the absence of it should be a clear indication that there is a mistake in the tagged output. For example, in the sentence (*The*, DET), (*glue*, NOUN), (*sets*, NOUN), (*in*, ADP), (*five*, NUM), (*minutes*, NOUN), it is clear that *sets* is a verb, even though it can also be used as a noun in other contexts. The taggers do not consider the grammatical correctness of the tagged sentences as a whole.

The DNN tagger predicts more nouns and verbs correctly than the other taggers, as well as adjectives when compared to Perceptron, Hunpos, and Brill. However, comparing its results to the Stanford tagger, the performance of the DNN tagger is not much better for the remaining classes (ADP, ADV, CONJ, PRON). In fact, the Stanford tagger correctly predicts more adverbs than the DNN tagger does. It appears that the DNN tagger often fails in similar circumstances as the other four taggers, as we have illustrated in the examples in this section.

6.2 Ongoing Work

In current work, we are considering a corrective approach for improving tagger performance, where sentences are first tagged by one of the taggers considered in this paper. In that process, certain words and patterns (described below) are identified for further analysis, which is carried out by applying a set of rules. For example, short, common words such as *away*, *as*, *because*, and so on, for which the tag assignment (typically ADJ, ADP, ADV, or CONJ) is often difficult and sometimes cannot be inferred from surrounding words, would be flagged and then checked against the rules. If any rule fits, the tag prescribed by the rule would be assigned. If not, the tag assigned by the original tagger would be kept. In some cases, the rules could involve more generic patterns and associated tag assignments. Consider, for example, the phrase … *a lousy quack !* (see Table 2). In general, the word *quack* is either a noun or a verb, and here evidently a noun. This case would be handled correctly by a rule such as DET ADJ {NOUN or VERB} ⇒ DET ADJ NOUN, where, in this case, the rule should only be applied at the *end* of a sentence. The curly brackets indicate that, from dictionary information (rather than just training data), the only *possible* tags for the last word are as listed. Similarly, this rule would also handle phrases such as … *a bitter harvest* (if it ends a sentence), where *harvest* could be a verb or a noun, but in this case is a noun.

7 CONCLUSION

In this paper, we have introduced a novel, challenging test set for POS tagging, with a single tagged word per sentence. In the development of the data set, we deliberately chose cases where at least one of four standard benchmark POS taggers fails to assign correct POS tags. We then applied a state-of-the-art DNN-based POS tagger to our data set, for a true out-of-sample test, and found a considerable drop in accuracy, from around $0.95 - 0.97$ over standard POS data sets (in line with reported values in the literature) to around 0.87 over our set, thus illustrating that POS tagging still presents significant challenges. Importantly, in our new data set, we explicitly removed ambiguous cases, so that linguistic ambiguity cannot be applied as an explanation for tagger failure. Indeed, as our analysis shows, we find many cases where the POS tagging is quite straightforward, but where both the four benchmark taggers, and the DNN-based tagger (albeit to a lesser degree), nevertheless fail to assign a correct tag.

REFERENCES

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Brants, T. (2000). Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, page 224–231, USA. Association for Computational Linguistics.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proc. of the Third Conference on Applied Natural Language Processing*, ANLC '92, page 152–155, USA. Association for Computational Linguistics.

Chiche, A. and Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):1–25.

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)*, pages 1–8.

Francis, W. N. and Kucera, H. (1979). Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Halácsy, P., Kornai, A., and Oravecz, C. (2007). Poster paper: HunPos – an open source trigram tagger. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212, Prague, Czech Republic. Association for Computational Linguistics.

Ling, W., Luís, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., Black, A. W., and Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096*.

Liu, S. and Ritter, A. (2022). Do CoNLL-2003 Named Entity Taggers Still Work Well in 2023? *arXiv preprint arXiv:2212.09747*.

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing*, pages 171–189, Berlin, Heidelberg. Springer Berlin Heidelberg.

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A., and Taylor, A. (1999). Treebank-3. *Linguistic Data Consortium, Philadelphia*.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Ramshaw, L. A. and Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Wu, Z., Deshmukh, A. A., Wu, Y., Lin, J., and Mou, L. (2023). Unsupervised Chunking with Hierarchical RNN. *arXiv preprint arXiv:2309.04919*.

Yasunaga, M., Kasai, J., and Radev, D. (2017). Robust multilingual part-of-speech tagging via adversarial training. *arXiv preprint arXiv:1711.04903*.