Reliable Part-of-Speech Tagging of Historical Corpora through Set-Valued Prediction

Stefan Heid[1]**, Marcel Wever**[1]**, Eyke Hüllermeier**[1]

[1]Heinz Nixdorf Institut, Paderborn University, Germany

Corresponding author: Stefan Heid , `stefan.heid@upb.de`

Abstract

Syntactic annotation of corpora in the form of part-of-speech (POS) tags is a key requirement for both linguistic research and subsequent automated natural language processing (NLP) tasks. This problem is commonly tackled using machine learning methods, i.e., by training a POS tagger on a sufficiently large corpus of labeled data. While the problem of POS tagging can essentially be considered as solved for modern languages, historical corpora turn out to be much more difficult, especially due to the lack of native speakers and sparsity of training data. Moreover, most texts have no sentences as we know them today, nor a common orthography. These irregularities render the task of automated POS tagging more difficult and error-prone. Under these circumstances, instead of forcing the POS tagger to predict and commit to a single tag, it should be enabled to express its uncertainty. In this paper, we consider POS tagging within the framework of set-valued prediction, which allows the POS tagger to express its uncertainty via predicting a set of candidate POS tags instead of guessing a single one. The goal is to guarantee a high confidence that the correct POS tag is included while keeping the number of candidates small. In our experimental study, we find that extending state-of-the-art POS taggers to set-valued prediction yields more precise and robust taggings, especially for unknown words, i.e., words not occurring in the training data.

Keywords

Part-of-speech tagging, set-valued prediction, uncertainty, historic corpora

I INTRODUCTION

In part-of-speech (POS) tagging, words of a corpus are assigned word classes, also referred to as POS tags. These tags form the basis for both automated natural language processing (NLP) procedures and for human experts who want to analyze texts or text passages more closely. Especially in studies devoted to the development of languages as a whole, its grammar and lexic, for which mainly historical texts are considered, POS tagging constitutes an elementary technique that is commonly applied as a first step, providing the basis for any subsequent analysis.

For modern languages such as English, state-of-the-art POS tagging algorithms based on machine learning methodology reach an accuracy as high as 97.96% [1], so that the task can essentially be considered as solved. Historic languages, on the other side, still impose challenges for POS tagging algorithms as well as human experts. The reasons for this are manifold:

- For such languages, the amount of data is very limited, and even the data available is comparatively noisy, because the labels are provided by humans who are not native speakers of that language.
- In addition to noisy data, a POS tagger needs to deal with further characteristics, such as missing regulations for orthography, and a multitude of word classes [8], i.e., distinct

labels.

- The right POS tag not only depends on the considered word but also on its role within a syntactical construction, and thus on the context in which it is used. However, in the investigated corpus, the context often cannot be easily differentiated due to a lack of syntactic markers, denoting the end of a main clause or a subordinated clause.
- Last but not least, there is a risk of projecting the understanding of contemporary speech onto the language under consideration, which is referred to as "comparative fallacy". This can lead to more systematic errors.

As a consequence, even experts are not entirely certain about many decisions. In [14], the inter-annotator agreement for a historic corpus of Middle Lower German (MLG) was reported at 92%, suggesting that there is significant room for different interpretations. Even though such numbers vary largely among datasets agreements as high as 98.6% have been reported for contemporary corpora[2]. Nevertheless, words are commonly labeled with single POS tags to keep the annotations consistent, in spite of a high risk to introduce mistakes. As the predicted POS tags are used in many up-stream NLP tasks, the reliability of the tagging is crucial to avoid error propagation. By requiring POS taggers to commit only to a specific tag, errors are deliberately enforced.

In this paper, we generalize the standard approach to POS tagging so as to enable a tagger to express its uncertainty, thereby making POS tagging more reliable. To this end, we build on the framework of set-valued prediction, which has attracted increasing attention in machine learning research in recent years [19], and integrate this framework with existing state-of-the-art POS taggers. More precisely, for a given word and its context, we allow a POS tagger to predict a set of candidate tags, for which it is sufficiently confident that the ground truth tag is contained in that set. Figure 1 presents a comparison of an exemplary tagging with classical and set-valued predictions.

Dat	is	vredebrake	Dat	is	vredebrake.
DDS	VAFIN	NA	{DDS}	{VAFIN, VKFIN}	{NA}

Figure 1: Example sentence taken from the MLG corpus (translation: this is breach of peace) tagged classically on the left-hand side and with set-valued prediction on the right. In this case it is uncertain, whether the verb *is* represents a finite auxiliary verb (VAFIN), as it is frequently the case, or a finite copular verb (VKFIN).

Set-valued tagging does not only increase reliability, but also comes with practical advantages. For example, the explicit handling of tagger uncertainty can help reduce a possible propagation of errors through a pipeline of NLP tasks, thereby increasing the reliability of down-stream tasks as well. Furthermore, it reduces the workload of a human annotator in charge of correcting the automatic annotations of a text, because it immediately points to those cases that are unclear and offers a reasonable number of candidates as options. This could be specifically useful for historic languages such as MLG, where POS-taggers are often used to speed up the annotation process. Yet, due to the limited amount of training data, the automatic annotation quality is relatively low in early stages.

In an empirical study, we demonstrate set-valued predictions to prove beneficial at the example of a historic corpus of legal texts written in MLG and a historical tag set comprising more than 90 word classes [8]. Furthermore, we find that the approach increases robustness of the POS-taggers in general and specifically for unknown words, i.e., words that did not occur in the training data.

The remainder of the paper is structured as follows. In the next section, we start with a brief overview of related work and the state of the art in POS-tagging, prior to introducing the problem

more formally in Section III. Our approach of set-valued prediction for part-of-speech tagging is then presented in Section IV and evaluated empirically in Section V. We conclude the paper with a summary and an outlook on future work in Section VI.

II RELATED WORK

Pos-tagging is generally done with a maximum entropy Markov model (MEMM) to exploit the sequential nature of the data. A MEMM is a generative model, where a base classifier predicts the tag probability given the word and its context. The choice of such a base classifier and the specific data representation and provided context information vary across different approaches. One of the earlier algorithms related to historic languages is the TreeTagger [24], which employs a decision tree as base classifier. It was first introduced with English language models and later on adapted to German [23] and it is still used in current research.

Ratnaparkhi [21] suggest a similar method but rely on logistic regression as a base learner. Their work has been further refined by McCallum, Freitag, and Pereira [17], who model the state transitions in a different manner. Lafferty, McCallum, and Pereira [15] extend the latter, solving an issue they refer to as *label bias problem*. The proposed algorithm is called conditional random fields (CRF).

Building on the work by Ratnaparkhi [21], Toutanova and Manning [29] refine MEMM models with a focus on unknown words. Toutanova et al. [30] extend this work further by incorporating the findings of Lafferty, McCallum, and Pereira [15]. However, their suggestion to solve the label bias problem is computationally less expensive. Instead of calculating a global model for state transitions, they build a MEMM which conditions on both previous and succeeding tags, resulting in the coreNLP tagger algorithm. For a more detailed discussion of the algorithm, we refer the interested reader to [11, ch. 5.4].

Deviating from the previously outlined approaches, Brants [3] shows that plain hidden Markov model (HMM)s are similarly powerful when used with trigrams and suitable statistical smoothing. Shen, Satta, and Joshi [28] propose an entirely different approach using perceptrons. Their algorithm slightly outperforms coreNLP. Collobert et al. [5] investigate recent neural network approaches for pos-tagging and other NLP-tasks. Huang, Xu, and Yu [12] incorporate the bidirectional nature of CRF with long short-term memory (LSTM), a specific type of recurrent neural networks, to cope with the sequential nature of the data.

The problem of pos-tagging is generally well-studied for contemporary English. However, it is often challenging if the data is noisy. For example, extensions to coreNLP have been proposed to enhance its performance when applied to short messages on twitter [7]. The noise in this data is mainly caused by large amount of unknown words, as new words emerge rapidly in the daily use of languages.

The similar temporal drift can be observed in historic corpora as they often span multiple centuries. This is often worsened by the sparsity of the data, particularly for historic German. Despite the fact that more potent taggers have been proposed in the recent past, TreeTagger is still used in linguistic research of historic German [10]. Also, CRF-based taggers have been used on MLG texts [14]. Recently [16] have employed deep learning techniques to historic datasets. Their work is, however, focused on lemmatization, which is often considered a prerequisite of pos-tagging.

III PROBLEM DEFINITION

In POS-tagging, the task is to assign each word (aka. token) w_i of a document, i.e., a sequence of words $D = (w_1, \ldots, w_n)$, a part-of-speech tag t_i. Finally, the whole document is thus assigned a sequence of POS-tags $\mathbf{t} = (t_1, \ldots, t_n)$. To put it formally, we are interested in predicting the probability of a tag t_i under the condition that we see word w_i:

$$P(t_i \mid w_i). \tag{1}$$

However, the part-of-speech tag depends not only on the word w_i itself, but also on the context in which it is used. In modern languages, the scope of the context is quite obvious, as it usually only encompasses the sentence in which the word occurs. While these sentences in modern language can be clearly distinguished by punctuation, historic languages typically lack such clear indications and the concept of (complex) sentences as we know them today. Still, the context is crucial for determining the role of a word within a construction, and therewith its part-of-speech tag. With context(w_i) a function referring to the context of the word w_i, we actually seek to estimate the probability of a tag t_i under the condition of the word w_i occurring within the context context(w_i):

$$P\big(t_i \mid w_i \wedge \texttt{context}(w_i)\big). \tag{2}$$

Let $\mathcal{W}$ be a vocabulary, $\mathcal{T}$ a tag set of size $s := |\mathcal{T}|$, and a context function, mapping the context of a word to a real-valued vector $\mathbb{R}^n$. We aim to learn a function $f : \mathcal{W} \times \mathbb{R}^n \to \mathcal{T}$, where $\mathbb{R}^n$ denotes the context description of the respective word. As we are also interested in the (un)certainty in the prediction of tag t_i for word w_i, we adapt the image of the function f to be a vector $\mathbf{p} = (p_1, \ldots, p_s) \in [0,1]^s$, where p_i correspond to the probability of tag t_i. Obviously, the entries should sum to 1, i.e., $\sum_{i=1}^s p_i = 1$.

The function is induced from training data of the form $(\mathcal{W} \times \mathcal{T})^l$ representing a document D for which each element w_i is labeled with a tag t_i. Here, l is the number of words of the particular document. An important property is the sequential order of the training data, where for $k \geq 1$ the elements w_{i-k}, $k \geq 1$ denote the preceding and w_{i+k} the subsequent words of w_i. The context function represents the context of a given word w_i as provided by these preceding and subsequent words and tags. The concrete instantiation of this function highly depends on the respective method and has no common signature. However, one simple example of such an instance is a binary encoding indicating whether a specific word is present or absent in the context.

IV SET-VALUED PREDICTION FOR PART-OF-SPEECH TAGGING

In standard supervised learning, the goal is to induce a predictive model $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the so-called instance space. To this end, the learner is given access to a set of training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^l \subset (\mathcal{X} \times \mathcal{Y})^l$. In multi-class classification, the output space $\mathcal{Y} = \{y_1, \ldots, y_s\}$ consists of a finite number of class labels (which correspond to the tags t_i in POS-tagging, where $\mathcal{Y} = \mathcal{T}$). Thus, a model h (hypothetically) assigns a class label $\hat{y} = h(\boldsymbol{x}) \in \mathcal{Y}$ to each query instance $\boldsymbol{x} \in \mathcal{X}$. The quality of a predictor is measured in terms of its expected loss $\mathbb{E}[\ell(y, h(\boldsymbol{x}))]$, where $\ell : \mathcal{Y}^2 \to [0,1]$ is a loss function that compares a prediction $\hat{y} = h(\boldsymbol{x})$ with a true outcome y, and the expectation is taken with respect to an underlying (though unknown) joint probability distribution on $\mathcal{X} \times \mathcal{Y}$.

4.1 Set-valued Prediction

Set-valued prediction is a generalization of the above setting, in which predictions $h(\boldsymbol{x})$ are subsets of $\mathcal{Y}$, i.e., predictors are now functions of the form $\mathcal{X} \to 2^{\mathcal{Y}}$. The basic idea is to

make "reliable" predictions $\hat{Y} = h(\boldsymbol{x})$ that cover the true class label y with high probability. At the same time, $\hat{Y}$ should of course not be too large, as a prediction would otherwise loose its value ($\hat{Y} = \mathcal{Y}$ is not very useful, despite being correct with probability 1). These two criteria, correctness and precision, are obviously in conflict with each other. Thus, in one way or the other, a predictor needs to find a reasonable compromise. Theoretically, there are different ways to learn such set-valued predictions. Mortier et al. [19] introduce algorithms that can calculate set-valued predictions for any probabilistic classifier, based on the predicted class distribution for a given data sample. To this end, they propose to generalize the notion of a loss function (or, equivalently, utility function). More specifically, they propose a class of utility functions of the following form:

$$u(y, \hat{Y}) = \underbrace{[\![y \in \hat{Y}]\!]}_{correctness} \cdot \underbrace{g(|\hat{Y}|)}_{precision}, \tag{3}$$

where $[\![\cdot]\!]$ denotes the indicator function. A utility function of that kind compares a ground truth class label y with a predicted set of candidates $\hat{Y}$. The utility is 0 in the case where $y \notin \hat{Y}$, i.e., if the true label is not covered by the prediction. Otherwise, the utility depends on the cardinality of the predicted set: g is a monotone decreasing function such that $g(1) = 1$, which means that the highest utility of 1 is reached if $\hat{Y} = \{y\}$ is a singleton set with the true label as its only element, and the utility decreases by adding further candidates. An example of a utility measure, which will be used in our experimental study later on, is the following function:

$$u_\beta(y, \hat{Y}) = \begin{cases} 0 & \text{if } y \notin \hat{Y}, \\ 1 - \left(\frac{|\hat{Y}|-1}{|\mathcal{Y}|-1}\right)^\beta & \text{if } y \in \hat{Y}, \end{cases} \tag{4}$$

This function, if parameterized by $\beta \in]0, \infty]$, defines how quickly the utility decreases when the size of the predicted set becomes larger.

Given a utility function (3) and a (conditional) probability $P(\cdot \mid \boldsymbol{x})$, or at least a prediction of that probability, a rationale prediction is a set Y^* that maximizes the expected utility

$$\begin{aligned} \mathbb{E}[u(\cdot, \hat{Y})] &= \sum_{y \in \mathcal{Y}} P(y \mid \boldsymbol{x}) \cdot u(y, \hat{Y}) \\ &= \sum_{y \in \mathcal{Y}} P(y \mid \boldsymbol{x}) \cdot [\![y \in \hat{Y}]\!] \cdot g(|\hat{Y}|) \\ &= g(|\hat{Y}|) \cdot \sum_{y \in \hat{Y}} P(y \mid \boldsymbol{x}) \,. \end{aligned}$$

In machine learning, a prediction of that kind is referred to as a Bayes-optimal prediction. To find this prediction efficiently, Mortier et al. [19] propose the Unrestricted Bayes-Optimal Prediction (UBOP) algorithm. Roughly speaking, this algorithms sorts the class labels y_i in decreasing order of their posterior probability $P(y_i \mid \boldsymbol{x})$ and successively computes the expected utilities of the top-k sets, i.e., the sets consisting of the k labels with highest probability. Starting with $k = 1$, the algorithms stops as soon as the expected utility decreases when adding another label. Under certain technical (though not very restrictive) assumptions on the function g, it can be shown that the set found in this way is indeed a Bayes-optimal prediction.

The approach by Mortier et al. [19] generalizes several other approaches to set-valued prediction, which are recovered as special cases for a suitable choice of the utility function. For example,
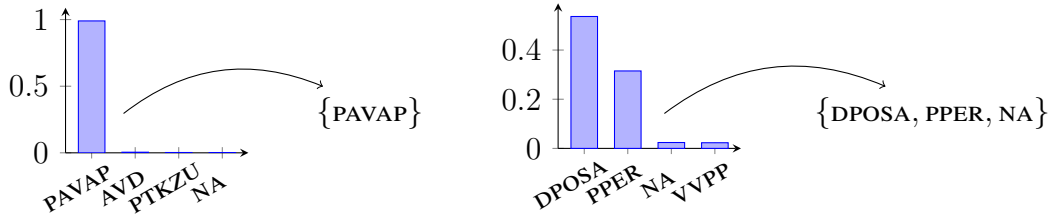
Figure 2: Visualization of UBOP. The posterior distribution over tags is converted into a set-valued prediction.

Del Coz, Díez, and Bahamonde [6] evaluate set-valued predictions in terms of utility scores from the information retrieval community, such as precision, recall, and the F_1-measure. Other researchers call the same setting *credal* or *cautious classification*. In a series of papers, they analyze several set-based utility scores that reward abstention in cases of uncertainty [32, 31]. The framework of *conformal prediction* also produces set-valued predictions, albeit with a focus on confidence (the predicted set covers the true class with a predefined probability) and less on utility [27]. As the use of the utility function framework results in Bayes-optimal predictions it allows to resolve the trade-off between large prediction sets and high confidence predictions in a more flexible way. Furthermore, set-valued prediction can be seen as a generalization of multi-class classification with a reject option [4, 20], where one either predicts a single class or the complete set of classes.

4.2 POS-Tagging

The above framework of set-valued prediction can also be applied to POS-tagging. In this case, the set of class labels $\mathcal{Y}$ is given by the tag set $\mathcal{T}$, and an instance is a word w_i together with its context. Moreover, the utility function is of the form

$$u(t, \hat{T}) = [\![t \in \hat{T}]\!] \cdot g(|\hat{T}|) , \tag{5}$$

and the expected utility of a subset of tags $\hat{T}$ is given by

$$\mathbb{E}[u(\cdot, \hat{T})] = \sum_{t \in \hat{T}} P(t \mid w_i \wedge \mathtt{context}(w_i)) \cdot g(|\hat{T}|) .$$

Figure 2 illustrates the application of the UBOP algorithm, i.e., the conversion from posterior probabilities to the set-valued predictions.

Recall that UBOP requires the posterior probability distribution (2) as input, and that the quality of this distribution will clearly have a strong impact on the quality of the set-valued prediction. In principle, probability estimates can be obtained by any probabilistic tagger whose context only depends on surrounding words but not on tags. The TreeTagger, for example, estimates probabilities in terms of relative frequencies of observations, which leads to relatively coarse predictions.

As CORENLP uses logistic regression, its probability estimates are better suited for set-valued prediction. However, since the performance of POS-taggers strongly relies on the contextual information harnessed, CORENLP also includes tags in the context. During prediction, this creates circular dependencies, as predicting a tag for a specific word expects the contextual tags to be set already. CORENLP introduces a strategy to circumvent these circular dependencies, which is based on a brute-force search stepping over (a subset of) all possible contexts. When applying

this approach to set-valued targets, the contexts blow up exponentially, making the problem intractable. Therefore, an approximation via post-processing is used in this work. To this end, the document is first tagged in a standard way to initialize the context information of each word, and then the tagging of every word is revised by a set-valued prediction using the context information obtained from the first pass. The pre-calculated context together with the word in consideration is used to build a feature vector allowing to cast the original sequential problem as a standard multi-class classification problem. In this way, we enable the application of the UBOP algorithm.

V   EXPERIMENTAL EVALUATION

We evaluate the proposed set-valued prediction approach to part-of-speech tagging, comparing two different types of POS-taggers, and relate their performance to a baseline, which is introduced in Section 5.3. To this end, we compare the performances obtained by the vanilla versions with the respective extensions producing set-valued predictions. Furthermore, we assess the generalization behavior of the approaches as well as their robustness for different amounts of training data.

5.1   Experimental Setup

In our experimental evaluation, we distinguish 3 scenarios for which we assess the generalization performance of the POS-taggers. To this end, we consider a set of N documents $\mathcal{D}^{(j)} = \{(x_i^j, y_i^j)\}_{i=1}^{l(j)}, 1 \leq j \leq N$. First, we split each document $\mathcal{D}^{(j)}$ once into 80% training and 20% test data, by cutting out a 20% fraction of the document. More precisely, we choose a point $c, 1 \leq c \leq \lfloor 0.8\, l(j) \rfloor$ and to obtain test data $\mathcal{D}_{\text{test}}^{(j)} = \{(x_i^j, y_i^j)\}_c^{c+\lfloor 0.2\, l(j) \rfloor}$ and training data $\mathcal{D}_{\text{train}}^{(j)} = \mathcal{D}^{(j)} \setminus \mathcal{D}_{\text{test}}^{(j)}$. Note that, independent of the scenario, the test data portion is never used for training, even if the approaches are not evaluated on that specific portion of the data.

The three scenarios are defined as follows[1].

Scenario 1 We evaluate the approaches for each document $D^{(j)}$ individually. To this end each method is trained on $D_{\text{train}}^{(j)}$ and evaluated on $D_{\text{test}}^{(j)}$. We refer to this scenario as *in-domain* performance.

Scenario 2 We evaluate the approaches on a leave-one-document-out fashion in order to assess the ability of the models being transferred to unseen texts, using the training data of all but one documents. More specifically, when assessing the performance for document k, we use $D_{\text{train}}^{(j)}\ \forall j \in [N] \setminus \{k\}$ and apply the methods on $D_{\text{test}}^{(k)}$.

Scenario 3 In the last and third scenario, we train the approaches on the entire training data $D_{\text{train}}^{(j)}\ \forall j \in [N]$ and subsequently apply the models to all test data portions $\mathcal{D}_{\text{test}}^{(j)}\ \forall j \in [N]$.

Our evaluation is then carried out in two parts. In the first part we report the accuracy of the vanilla versions of the POS-taggers and the utility and the mean size of the predicted POS-tag sets of the set-valued prediction POS-taggers. The parameter β of the utility function (4) is set to 1. The second part of our evaluation concerns the sensitivity of the approaches with respect to the parameter choice of β. Varying the value of this parameter, which penalizes the prediction of larger sets, we investigate the influence on the size of the predicted sets.

All the experiments were run on a single machine equipped with 6 cores (i7-8700k for hyper-parameter optimization runs and i7-9750H for the evaluation of the POS-taggers) and 32GB of RAM. The machine operates with a Manjaro Linux 5.4.39, Python 3.8 and Java 10.

---

[1]We use $[N]$ as a more convenient notation for referring to the set $\{1, 2, \ldots, N\}$.

5.2 Corpus Description

The corpus considered in this study mainly stems from the InterGramm project[2] [18, 25, 26] and consists of dateable and localizable legal documents and records of judgements ranging from the 13th to the 17th century comprising 23 documents in total. Some texts of dataset originally stem from the ReN project [22], which have been partially adapted to be consistent to the tagging guidelines of the InterGramm project. The majority of the documents originate from the Central Low German region written in a historic German dialect called Middle Low German. The remaining corpus contains the first Early New High German texts that were written in Low German after the change of writing language. In the course of the InterGramm project, these texts have been annotated with POS-tags which are considered as ground truth labels.

Since an established, commonly accepted orthography did not exist at that time, we unified different spellings of the same words to condense the number of different words in a reasonable way. To this end, we first applied a clustering algorithm using the Levenshtein distance as the distance metric and let an expert double-check the correctness of the discovered clusters. As an example we would substitute *denen*, *denet*, and *dhenet* by *denen*. The data used in this study are provided (together with the code to read and process the documents) on GitHub[3].

For the sake of clarity, the evaluation presented in Section 5.4 focuses on four selected documents from the Corpus, which are representatives of different challenges when dealing with historic corpora. The results for the remaining corpus are provided together with the code via the GitHub repository. The *Duisburger Stadtrecht (1500)* can be seen as an outlier document, as the dialect in this text is quite different from the rest of the corpus. *Bremer Urkunden (1351–1400)* on the other hand can be seen as the medoid of the data set, as the corpus contains many other deeds from the same time, some of which are even from Bremen as well. *Bambergische Halsgerichtsordnung (1510)* is one of the most recent documents in the corpus. Finally, we consider the *Kolberger Kodex (1297)*, which in turn is a rather old one. In the following, the four documents will be referred to as Duisburg, Bremen, Bamberg, and Kołobrzeg.

5.3 Baseline

To compare the different algorithms on a common ground, a baseline is used to estimate the difficulty of the data set itself. The baseline predicts the tag with the highest probability and solely depends on the word itself. In case the word w_i is known, the predictor is as follows:

$$t_i = \arg\max_{t \in \mathcal{T}} P(t \mid w_i)$$

This posterior probability is estimated by the relative word frequency:

$$P(t \mid w) = \frac{\#(t, w)}{\#(w)}$$

If the word is unknown, the baseline predicts the globally most frequent tag via the prior:

$$t_i = \arg\max_{t \in \mathcal{T}} P(t)$$

In our corpus, this is the NA tag, which represents a common noun. Aligning to the other algorithms, this tagger can only predict tags that are known from the training data.

Within the set-valued prediction framework, the conditional probability distributions, or the prior in case of an unknown word, are used directly.
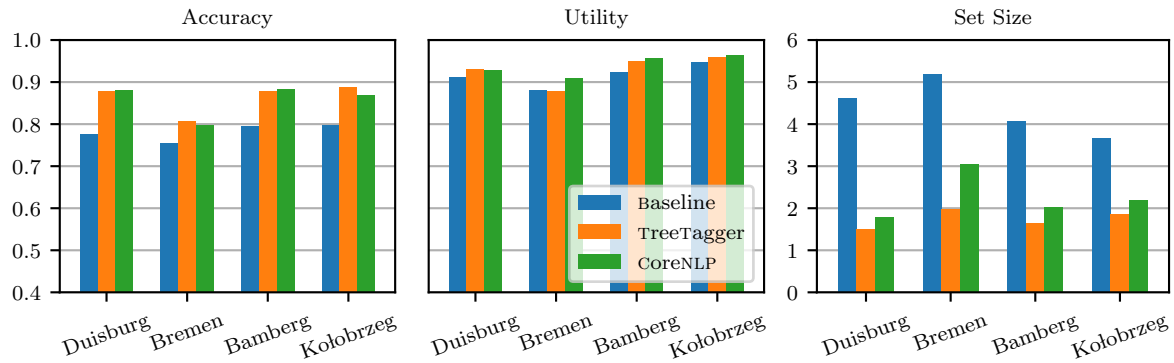
---

[2] https://www.uni-paderborn.de/forschungsprojekte/intergramm
[3] https://github.com/stheid/SetPOS

Figure 3: Performance numbers of the different taggers when trained and evaluated on one document.

5.4 Results

Across all experiments, the TreeTagger and CORENLP showed competitive results. Generally speaking, the TreeTagger predicts smaller sets with slightly inferior accuracy and utility compared to CORENLP. The weaker performance was generally more consistent for the utility measure. The baseline was outperformed by both taggers in nearly every scenario and measure. The margin was generally larger when assessing the accuracy, while performances with respect to utility were more similar.

The results of the in-domain scenario, where the tagger was trained and evaluated on one document, is shown in Figure 3. In this scenario, the data is as homogeneous as possible, while providing much less training data than in the other scenarios. The accuracy on the smaller documents such as Bremen (7.5 k tokens) and Kołobrzeg (13.2 k tokens) indicates that the TreeTagger is more data efficient than CORENLP. With the Bremen document being particularly small, all taggers including the baseline make more than one tagging error in five words. For the larger texts Duisburg (15.5 k tokens) and Bamberg (19.7 k tokens), CORENLP provides the best accuracy.

When comparing the accuracy with the utility, all taggers recover a lot of their errors, as they are allowed to provide more tags on the small Bremen document. Simultaneously, the set sizes for the Bremen document are the largest, regardless of the tagger. However, they are only slightly larger than for the other documents. Specifically the TreeTagger has only a slightly increased set size, but also performs poorer than the baseline in terms of utility.
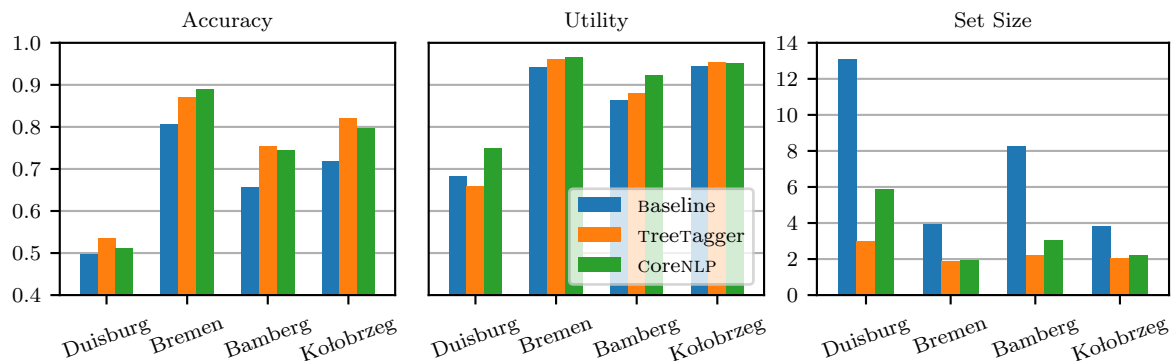

Figure 4: Performance numbers of the different taggers when trained and evaluated in a leave-one-document-out fashion.

Figure 4 shows the performance on the transferability scenario. Since Duisburg has a different

dialect than the other documents, none of the classifiers is able to predict more than half of the tags correctly. The Bamberg and Kołobrzeg texts show similar performance drops, though less severely. For the Bremen document, the classifiers present superior accuracy as compared to the in-domain scenario. The larger data set allows thus for a higher performance than when training only on the homogeneous document itself.

Regardless of the text or approach, the utility measures are significantly higher. For a difficult text such as Duisburg, the performance increase over the accuracy comes at the cost of an increased set size. However, in a real-world scenario, i.e., in interaction with a human annotator, such a suggestion might still be more useful than predicting a wrong tag half of the time.
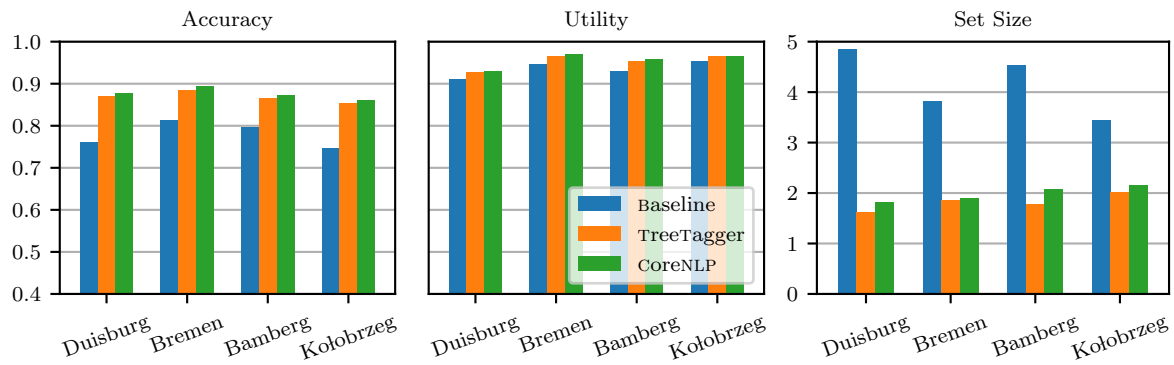

Figure 5: Performance numbers of the different taggers when trained on the whole corpus.

Finally, Figure 5 shows the results when using the training section of the entire corpus. Providing the classifier with additional training data as compared to the in-domain scenario can also result in detrimental effects. For example, the prediction performance for the Bremen document can be negatively affected by adding portions of the Duisburg text to the training data. The performance on the Kołobrzeg document suffers significantly in accuracy as compared to the first scenario.

However, it is a non-trivial task to decide which data to exclude from the training data by looking only at unlabeled evaluation data. The results of the in-domain scenario and training on the whole corpus are comparable, however, meaning that the classifiers are not severely affected by confusing data. Again, the utility proves to be more robust in terms of performance. Even though the accuracy on the Kołobrzeg text dropped when considering the whole corpus, the utility is not affected negatively at all. For TreeTagger and coreNLP, the set size across all data sets is on average around 2, clearly outperforming the baseline.

In this scenario, coreNLP outperforms the TreeTagger in accuracy and utility, whereas the set sizes of the TreeTagger are typically smaller. However, if it is desirable to have smaller set sizes, this can also be enforced in the parameterization of the UBOP algorithm to force coreNLP to return smaller sets as well. Both taggers use the optimal set size for a specific posterior and penalty function.

5.5 Detailed Analysis
In this section, we analyze the sensitivity of the results with respect to the parameter β of the utility function (Equation 4). Recall that this parameter defines the amount of risk aversion, as it modifies the penalty incurred by larger set sizes. As $\beta \to 0$, the classifier predicts only singleton sets. For $\beta \to \infty$, the classifier behaves more risk averse, preferring larger sets to ensure the correct tag to be covered with a higher probability.

The results shown in Figure 6 are obtained by averaging all evaluations of the third scenario, where the classifier is trained on the whole corpus. Notably, the utility increases with larger values of β. However, values larger than one barely increases the utility score. This is mainly because the utility can only increase if either the penalty for a large set size is decreased, or the recall is increased due to the increased coverage of the posterior distribution. When investigating the average set size, we see that the baseline always results in the largest sets and the TreeTagger provides the smallest sets.

For the baseline, this is in agreement with the confidence of the tagger, as it also performs the worst. However, the TreeTagger yields smaller set sizes while showing smaller confidence as compared to coreNLP. This indicates that the probability distributions obtained through the decision tree, which are estimated as fractions of observed frequencies of features in the training data, are less well calibrated. This is made especially apparent by the fact that even the baseline can outperform the TreeTagger when selecting $\beta > 2$. In contrast to this, the logistic regression model as employed by coreNLP tends to produce a more fine grained and better calibrated posterior.
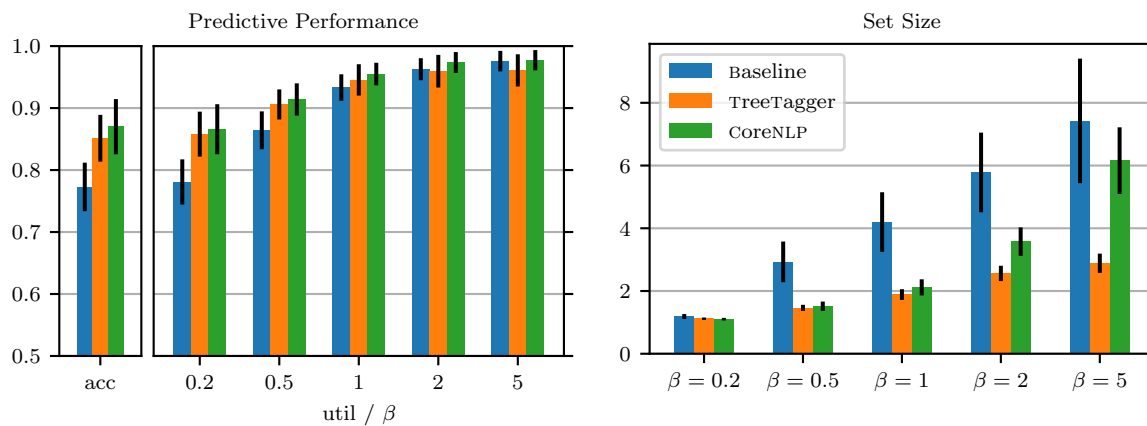

Figure 6: The average utility and average set size with respect to the parameter β. Larger values of β result in a more risk averse behaviour. The performance is aggregated first within each document and then averaged again. The error bars show the standard deviation of the values.

Finally, the distribution over the predicted set sizes is analyzed in Figure 7. For this experiment, coreNLP was trained on the training data of the whole corpus and evaluated on the complete test section with $\beta = 1$. For known words, the large majority of predictions are basically singletons. The distribution of the set sizes is decreasing almost exponentially. The histogram over unknown words decays much slower. Still, the majority of unknown word predictions results in a singleton prediction. Predictions of sets containing more than 8 elements are extremely seldom in any case.

VI CONCLUSION

We proposed an extension of POS-tagging, in which a tagger is allowed to provide set-valued annotations in cases of uncertainty. To this end, we leveraged methods for set-valued prediction as recently developed in the field of machine learning and combined them with state-of-the-art POS taggers. In our empirical study, in which we analyzed the specifically relevant case of annotating texts from a historic language, we observed an improved utility as compared to the standard POS-tagging. Another observation is that set-valued predictions seem to be more robust
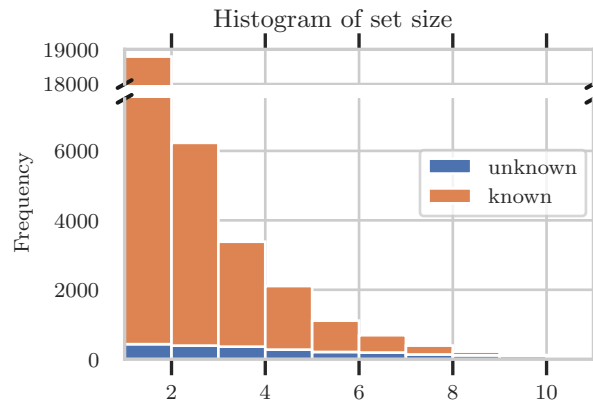
Figure 7: Distribution of the set sizes. The histogram is calculated from the predictions on the whole corpus with CORENLP fitted on the train portion of the whole data set

with regard to the employed base learner, while the set sizes (number of candidate tags) are still moderate on average.

In future work, we plan to elaborate on promising extensions of the approach to reliable tagging as presented in this paper. One idea, for example, is to take multiple potential contexts into account, i.e., different (structurally ambiguous) readings of a text passage. For example, the current extension of the CORENLP tagger is limited to the most likely context extracted via a first pass of the algorithm, thus completely ignoring less likely contexts. Another interesting idea is to combine set-valued prediction with set-valued supervision. Thus, instead of assuming deterministic annotations in the training data, the human expert would be allowed to provide set-valued annotations in cases where he or she feels uncertain and considers several tags as plausible options — just like the tagger at prediction time. Learning from "weak" supervision of that kind can be accomplished by means of techniques for so-called superset learning [13, 9].

REFERENCES

[1] Bernd Bohnet et al. "Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings". In: *arXiv preprint arXiv:1805.08237* (2018).

[2] T. Brants. "Inter-annotator Agreement for a German Newspaper Corpus". In: *LREC*. 2000.

[3] Thorsten Brants. "TnT: a statistical part-of-speech tagger". In: *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics. 2000, pp. 224–231.

[4] Chi-Keung Chow. "On optimum recognition error and reject tradeoff". In: *IEEE Transactions on information theory* 16.1 (1970), pp. 41–46.

[5] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.Aug (2011), pp. 2493–2537.

[6] Juan José Del Coz, Jorge Díez, and Antonio Bahamonde. "Learning Nondeterministic Classifiers". In: *The Journal of Machine Learning Research* 10 (2009), pp. 2273–2293.

[7] Leon Derczynski et al. "Twitter part-of-speech tagging for all: Overcoming sparse and noisy data". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. 2013, pp. 198–206.

[8] Stefanie Dipper et al. "HiTS: ein Tagset für historische Sprachstufen des Deutschen." In: *JLCL* 28.1 (2013), pp. 85–137.

[9] E. Hüllermeier and W. Cheng. "Superset Learning Based on Generalized Loss Minimization". In: *Proceedings ECML/PKDD–2015, European Conference on Machine Learning and Knowledge Discovery in Databases*. Porto, Portugal, 2015.

[10] Nora Echelmeyer, Nils Reiter, and Sarah Schulz. "Ein PoS-Tagger für das Mittelhochdeutsche". In: (2017).

[11] Stefan Heid. "Set-Valued Prediction for Part-of-Speech Tagging". MA thesis. Paderborn University. DOI: 10.17619/UNIPB/1-957.

[12] Zhiheng Huang, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging". In: (2015). arXiv: 1508.01991.

[13] E. Hüllermeier. "Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization". In: *International Journal of Approximate Reasoning* 55.7 (2014), pp. 1519–1534.

[14] Mariya Koleva et al. "An automatic part-of-speech tagger for Middle Low German". In: *International Journal of Corpus Linguistics* 22.1 (2017), pp. 107–140.

[15] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[16] Enrique Manjavacas, Ákos Kádár, and M. Kestemont. "Improving Lemmatization of Non-Standard Languages with Joint Learning". In: *NAACL-HLT*. 2019.

[17] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation." In: 2000.

[18] Marie-Luis Merten, Nina Seemann, and Marcel Dominik Wever. "Grammatikwandel digital-kulturwissenschaftlich erforscht. Mittelniederdeutscher Sprachausbau im interdisziplinären Zugriff". In: *Niederdeutsches Jahrbuch* 142 (2019), pp. 124–146.

[19] Thomas Mortier et al. "Efficient Algorithms for Set-Valued Prediction in Multi-Class Classification". In: (2019). arXiv: 1906.08129.

[20] Harish G. Ramaswamy, Ambuj Tewari, and Shivani Agarwal. "Consistent Algorithms for Multiclass Classification with a Reject Option". In: *CoRR* abs/1505.04137 (2015).

[21] Adwait Ratnaparkhi. "A maximum entropy model for part-of-speech tagging". In: *Conference on Empirical Methods in Natural Language Processing*. 1996.

[22] ReN-Team. *Referenzkorpus Mittelniederdeutsch/Niederrheinisch (1200-1650)*. http://hdl.handle.net/11022/0000-0007-D829-8. Version 1.0. Aug. 2019.

[23] Helmut Schmid. "Improvements in part-of-speech tagging with an application to German". In: *Natural language processing using very large corpora*. Springer, 1999, pp. 13–25.

[24] Helmut Schmid. "Probabilistic Part-of-Speech Tagging Using Decision Trees". In: *Conference on New Methods in Language Processing. Manchester, UK*. 1994.

[25] Nina Seemann et al. "Annotation Challenges for Reconstructing the Structural Elaboration of Middle Low German". In: *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Association for Computational Linguistics (ACL), 2017, pp. 40–45. DOI: 10.18653/v1/W17-2206.

[26] Nina Seemann et al. "Supporting the Cognitive Process in Annotation Tasks". In: *Postersession Computerlinguistik der 40. Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft*. Ed. by Kerstin Eckart and Dominik Editors Schlechtweg. 2018.

[27] G. Shafer and V. Vovk. "A Tutorial on Conformal Prediction". In: *Journal of Machine Learning Research* 9 (2008), pp. 371–421.

[28] Libin Shen, Giorgio Satta, and Aravind Joshi. "Guided learning for bidirectional sequence classification". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*. 2007, pp. 760–767.

[29] Kristina Toutanova and Christopher D Manning. "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger". In: *Proceedings of the 2000 Joint SIGDAT*

conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*. Association for Computational Linguistics. 2000, pp. 63–70.

[30]   Kristina Toutanova et al. "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for computational Linguistics. 2003, pp. 173–180.

[31]   G. Yang, S. Destercke, and M. Masson. "The Costs of Indeterminacy: How to Determine Them?" In: *IEEE Transactions on Cybernetics* 47 (2017), pp. 4316–4327.

[32]   Marco Zaffalon, Corani Giorgio, and Denis Deratani Mauá. "Evaluating credal classifiers by utility-discounted predictive accuracy". In: *Int. J. Approx. Reasoning* 53 (2012), pp. 1282–1301.