Received: 14 May 2021 | Revised: 24 March 2022 | Accepted: 19 August 2022
DOI: 10.1002/sam.11596

WILEY

RESEARCH ARTICLE

An auxiliary Part-of-Speech tagger for blog and microblog cyber-slang

Silvia Golia[1] | **Paola Zola**[2]

[1]Department of Economics and Management, University of Brescia, Brescia, Italy
[2]Istituto di Informatica e Telematica, CNR, Pisa, Italy

Correspondence
Silvia Golia, Department of Economics and Management, University of Brescia, Brescia, Italy.
Email: silvia.golia@unibs.it

Abstract
The increasing impact of Web 2.0 involves a growing usage of slang, abbreviations, and emphasized words, which limit the performance of traditional natural language processing models. The state-of-the-art Part-of-Speech (POS) taggers are often unable to assign a meaningful POS tag to all the words in a Web 2.0 text. To solve this limitation, we are proposing an auxiliary POS tagger that assigns the POS tag to a given token based on the information deriving from a sequence of preceding and following POS tags. The main advantage of the proposed auxiliary POS tagger is its ability to overcome the need of tokens' information since it only relies on the sequences of existing POS tags. This tagger is called auxiliary because it requires an initial POS tagging procedure that might be performed using online dictionaries (e.g., Wikidictionary) or other POS tagging algorithms. The auxiliary POS tagger relies on a Bayesian network that uses information about preceding and following POS tags. It was evaluated on the Brown Corpus, which is a general linguistics corpus, on the modern ARK dataset composed by Twitter messages, and on a corpus of manually labeled Web 2.0 data.

KEYWORDS
ARK dataset, Bayesian network, Brown corpus, computational linguistics, part-of-speech

1 | INTRODUCTION

Over the last few decades, technological progress in computational resources required a lot of effort to model large amounts of textual data. Moreover, in recent years, the Internet expansion, the Web 2.0 phenomenon, and massive mobile device adoption have changed communication languages. Therefore, earlier analyses in the field of natural language processing (NLP) carried out on documents written abiding by the standard language rules might not be efficient if applied to modern textual data. Web 2.0 text data, such as blogs and microblogs messages, lead to a wide range of applications in different research fields: event detection [1], sentiment analysis [2], political disclosure [3], and location inference [4]. Web 2.0 data, and specifically microblog data, are characterized by the absence of a standard vocabulary: slang, abbreviations, symbols, emoticons, and so on have replaced traditional words, creating the so-called *cyber-slang*. Moreover, Web 2.0 content contains grammar mistakes, typos, and emphasized words (e.g., "nooo"). Extracting information from Web 2.0 texts is a relatively new challenge for NLP. Traditional NLP methods, and especially state-of-the-art Part-of-Speech (POS) taggers, work well for structured text, but they have poor data tagging capabilities for blogs and microblogs data [5]. The main consequence is the inability of the state-of-the-art POS taggers to assign a meaningful POS tag to all the words in a Web 2.0 text [6]. Some taggers assign particular labels to unseen words, whereas other do not assign any tags at all. In order to overcome this limitation, this paper proposes an auxiliary POS tagger that assigns the POS tag to an unknown word based on the information deriving from a POS tags sequence. It intervenes after an initial POS tagging step of a corpus, predicting the remaining unknown POS tags that do not match any dictionary or for which the morpheme analysis is not helpful.

The auxiliary POS tagger works in two stages: first, it predicts the probability distribution of the unknown POS tag and, second, it summarizes it into a predicted POS tag. The probability distribution of the unknown POS tag is calculated using a Bayesian network (BN) built on a sequence of POS tags. The BN is a probabilistic graphical model that makes explicit, through a directed acyclic graph, the interactions within a set of variables [7, 8]. As for the second step, the paper considered two classifiers, in addition to the default Bayes Classifier, to summarize the above-mentioned distribution. The predictive performance of the learned BN was compared to the predictive performance of the random forests (RF) [9], which is a popular machine learning model. In this study, RF was used as a benchmark for the BN, given its high predictive power in the classification context [10]. The main advantage of the proposed method is its flexibility when applied to different domains (e.g., financial, journalistic, medical, etc.) and different kinds of texts (including traditional corpus, and modern blog and microblog data). Its flexibility is due to its ability to overcome the need for token knowledge. Moreover, the method can be applied to any language, after a preliminary estimation of the BN trained on a large tagged corpus. Another important difference between the proposed method and the existing POS taggers regards missing values; as soon as the BN is estimated, it can be used to predict the missing tag without needing to know the entire set of predictors used in the estimation step, thus, overcoming a pre-imputation stage commonly used in many machine learning (such as support vector machines [11]) and deep learning algorithms [12].

However, since the proposed method is auxiliary to an initial POS tag phase, it needs a partially labeled text to be applicable. This, in practical terms, translates to a need for an online dictionary or a POS tagging algorithm.

The paper is structured as follows. Section 2 reports an overview of the existing literature on POS tagging, whereas Section 3 describes the proposed method and the evaluation metrics used in the analysis. Section 4 reports the description of three experiments that helped evaluating the validity of the proposed methodology. Conclusions follow in Section 5.

2 | RELATED WORK

Earlier works on sentence POS tagging were mainly based on grammar rules and morphemes, such as one of the first large-scale systems called TAGGIT [13]. TAGGIT uses 71 different tags and a disambiguation grammar including 3300 rules, reaching an overall accuracy of around 77% of the words in the Brown University Corpus. Another rule-based approach was proposed by [14] for the *Wall Street Journal* (WSJ) dataset. With the progress in computational technology and the growing interest in machine learning models, POS studies evolved also in this respect. A wide range of research has been done since 1970, focusing mainly on Markov models (MM) [15] and their variants such as the conditional Markov models (CMM) [16] and the hidden Markov models (HMM) [17]. MM approaches are unidirectional methods computing the probability of a tag at time t when considering its preceding tag (at time $t-1$).

In the 90s, another important work on POS tagging was published by Schmid [18]. The author proposed the TreeTagger, an algorithm for POS tagging based on decision trees. The initial work was performed on English corpora but was later extended to several more languages and is today one of the most used and popular POS taggers. Toutanova et al. [19] introduced a model called cyclic dependency networks (CDN) that considers also following information and, together with the token lexical analysis, they proposed one of the most used and popular POS taggers: the Stanford log-linear. Over the last decade, further improvement in POS tagging investigation was achieved through the application of deep learning algorithms such as bidirectional long short-term memory neural network (BI-LSTM) on embedded text representation [20, 21]. Moreover, the latest developments in artificial intelligence studies encouraged the application of adversarial training (AT) [22] and autoencoders [23] for sequence POS tagging. Yasunaga et al. [22] analyzed the proposed AT model both on the English corpus (i.e., WSJ) and on the universal dependencies (UD) dataset with more than 20 different languages, while Zhang et al. [23] sampled eight languages from the same UD collection.

Recent studies tried to extend traditional POS taggers to blogs and microblogs data, yet with poor results. Conditional random fields (CRF) has been widely adopted for POS tagging, especially for Indian languages, such as Urdu [24, 25] and Bengali [26]. CRFs are undirected graphical models which are very similar to HMMs but with high computational complexity, making them less efficient than other algorithms [27]. Nevertheless, for the Urdu POS tagging task, Khan et al. [24] have shown that CRF performed better than support vector machine (SVM), two variants of the recurrent neural network (RNN) and HMM. Nand and Perera [5] compared different traditional POS taggers based on different approaches: maximum entropy (ME), HMM, and CRF. As training and test data, they used three different English Twitter datasets (T-POS, DCU, and ARK). The T-POS and the DCU dataset, together with the Penn Treebank section related to the WSJ and NPS IRC datasets, were also analyzed by Derczynski et al. [28], who evaluated four existing POS tag algorithms: Tnt [29], SVMTool [30], TBL [31], and Stanford log-linear tagger. The authors calculated different metrics on Twitter test data based on different training sets, varying the impact of Twitter data as opposed to the traditional corpus. The analysis showed the loss in accuracy when the models were trained on a structured corpus with a small portion of Twitter data, therefore, a new methodology based on the vote constrain bootstrapping (VCB) was proposed. Attardi and Simi [32] proposed a traditional algorithm (HMM) and a more recent BI-LSTM to estimate the POS tag of annotated Italian tweets. Albogamy and Ramsay [33] compared the POS taggers AMIRA, MADA, and Stanford log-linear on an Arabic sample of tweets. The authors showed a loss in accuracy in about 30%–40% of the tweets as opposed to traditional text, and they proposed a series of improvements for the compared POS taggers that were able to increase the goodness by about 15%.

Table 1 contains a summary of the main features of a representative subset of studies on POS tagging, where consolidated and advanced algorithms were applied to Web 2.0 data and/or traditional corpus data. The comparisons in Table 1 aim to highlight the differences in terms of datasets, algorithms, validation methods, and evaluation metrics. Moreover, column *Cross Domain* states if the study involved a domain adaptation approach, while column *Lexicon* refers to the usage of token's morphological analysis. Table 1 also shows, in its last row, how this study is described according to the features being considered. The main difference between the proposed method and the others reported in Table 1 is its ability to discard any lexicon information (such as suffixes, etc.). Moreover, even if the BN belongs to the class of the probabilistic graphical models [34] like HMM and CMM, it differs in some regards. Both HMM and CMM derive the tags sequence from the corresponding words sequence, using only the information related to preceding tokens and tags, while the proposed auxiliary POS tagger based on BN also employs the knowledge of the subsequent tags. Moreover, CMM and most machine learning models suffer in the testing procedure when some preceding tags are missing, whereas BN can estimate the target tag even if some of the preceding or subsequent tags are missing.

3 | PROPOSED METHOD

As stated in the introduction, Web 2.0 texts are a challenge for POS taggers because they include words that do not match any dictionary or that are written in non-standard ways (e.g., "im", "youuuu", "lovethat"). Some traditional POS taggers, therefore, attempt to predict the "unseen" words often assigning a residual POS tag class, such as "Foreign Word," while other POS taggers leave the token unlabelled.

Table 2 shows an example of a tweet which includes slang, hashtags, mentions, and emoticons. We used the Cambridge online dictionary[1] and the state-of-the-art Stanford log-linear POS tagger to tag the words. The "?" symbol indicates the words not found by the Cambridge online dictionary, whereas the "$_\times$" symbol for the Stanford log-linear POS tagger is the tag for foreign word (FW). Six words are cyber-slang tokens; five of them were recognized as unknown by the dictionary, whereas the Stanford log-linear POS tagger labeled four of them as FW. Curiously, the "luv" word, which is slang for the verb "to love," was wrongly tagged by the dictionary because luv is an actual word in English. Forty-seven percent of the words were tagged as FW by the Stanford log-linear POS tagger; this is an inexplicably high percentage suggesting that some of the corresponding words are typical cyber-slang words, therefore unknown.

Following the Twitter example in Table 2 and the limits of both existing POS taggers and online dictionaries, the idea is to focus on a POS tags sequence to derive the unknown POS tag without performing any word vectorization nor analyzing the word string (affixes, capital letters, etc.). Thus, the only available information that we aim to use for labelling a target tag is the sequence of preceding and subsequent POS tags determined by an initial POS tagging step (see Table 2). The auxiliary POS tagger first derives a suitable BN from a wide and tagged corpus (such as the Brown Corpus), allowing to predict the probability distribution of the unknown POS tag, and then it applies a criterion to summarize that distribution into a predicted POS tag.

Identifying the best BN requires considering several information sets to determine the most suitable tag sequence length to be used for predicting the unknown POS tag, denoted as tag_t. Most of the previous works only relied on the information linked to the two preceding POS

[1] https://dictionary.cambridge.org/

TABLE 1 Summary of the main features of a representative subset of studies on POS tagging

Study	Source[a]	Cross Domain	Dataset[b]	Language[c]	Algorithm[d]	Lexicon	Metrics[e]	Validation[f]
Cutting et al. [17]	TC	No	Brown	ENG	HMM	Yes	Acc	8 ITE
Schmid [18]	TC	No	PTB	ENG	DT	Yes	Acc	
Cussens [14]	TC	No	WSJ	ENG	Rules	Yes	Acc	
Ratnaparkhi [16]	TC	No	PTB	ENG	CMM	Yes	Acc	
Toutanova et al. [19]	TC	No	PTB	ENG	CDN	Yes	Acc	
Ekbal et al. [26]	TC	No	NLPAI, SPSAL	BAN	CRF	Yes	Acc	10-F Cv
Derczynski et al. [28]	TC, B	Yes	T-POS, DCU WSJ NPS IRC	ENG	VCB	Yes	Acc	HO, 10-F Cv
Albogamy and Ramsay [33]	B	No	Twitter	ARAB	IMP-POS	Yes	Acc	–
Nand and Perera [5]	B	Yes	T-POS DCU ARK	ENG	HMM CRF ME	Yes	Acc	3-F Cv
Attardi and Simi [32]	B	No	SENTICPOL Evalita 2009	IT	HMM BI-LSTM	Yes	Acc	–
Yasunaga et al. [22]	TC	No	WSJ, UD	ENG, MX	AT	Yes	Acc	HO
Zhang et al. [23]	TC	No	UD	MX	NCRF-AE	Yes	Acc	HO
Khan et al. [24]	TC	No	CLE, BJ	URDU	CRF HMM SVM RNN	No	Acc	10-F Cv
This study	TC, B	Yes	Brown ARK	ENG	BN	No	Acc, AUC, MAF1, Av Acc	10-F Cv

[a] *Source*: B, blog and microblogs; TC, traditional corpus.
[b] *Dataset*: BJ, Bushra Jawaid; CLE, Center for Language Engineering; PTB, Penn TreeBank; WSJ, Wall Street Journal; UD, Universal Depencencies.
[c] *Language*: ARAB: Arabic, BAN: Bangali, ENG, English; IT, Italian, MX, mixed languages.
[d] *Algorithm*: AT, adversarial training; BI-LSTM, bidirectional long- and short-term memory networks; BN, Bayesian network; CDN, cyclic dependency networks; CMM, conditional Markov Model; CRF, conditional random field; DT, decision trees; HMM, hidden Markov models; IMP-POS, improvement of ADMIRA, MADA, and Stanford Log-linear taggers; ME, maximum entropy; NCRF-AE, neural CRF autoencoder; RNN, recurrent neural network; Rules, grammatical rules; SVM, support vector machine; VCB, vote constrain bootstrapping.
[e] *Metrics*: Acc, accuracy; Av Acc, average accuracy; AUC, area under the ROC curve; MAF1, macro average F1-score; P, precision; R, recall.
[f] *Validation*: Cv, cross-validation; HO, held-out, ITE, iteration; n-F Cv, *n*-fold cross-validation.

tags; in our analysis, we investigated the three possible sets of information below:

- one tag before and one tag next ($Tag_{t-/+1}$): $\{tag_{t-1}, tag_t, tag_{t+1}\}$,
- two tags before and two tags next ($Tag_{t-/+2}$): $\{tag_{t-2}, tag_{t-1}, tag_t, tag_{t+1}, tag_{t+2}\}$,
- three tags before and three tags next ($Tag_{t-/+3}$): $\{tag_{t-3}, tag_{t-2}, tag_{t-1}, tag_t, tag_{t+1}, tag_{t+2}, tag_{t+3}\}$.

We did not consider extra tags preceding and following an unknown POS tag, such as tag_{t-4} or tag_{t+4}, because a longer sequence of predictors could cause two kinds of problems: first, an unknown POS tag might appear at the beginning or at the end of a sentence, in which case a longer set of preceding and following tags could include

TABLE 2 Example of a tweet tagged by Cambridge Dictionary and Stanford log-linear Part-of-Speech (POS) tagger

Token	Cambridge dictionary	Stanford POS tagger
@Iselgomezl	?	×
Dont	?	×
Really	ADV	ADV
Know	VERB	VERB
Where	ADV	ADV
I	PRON	×
Would	VERB	VERB
Be	VERB	VERB
Without	ADP	ADP
Youuuuuuuuuu	?	NOUN
And	CONJ	CONJ
Demi	NOUN	×
<3 (♥)	?	×
I	PRON	×
Luv	NOUN	×
Youu	?	×
#IfMyMomsHadATwitter	?	NOUN

Abbreviations: ADP, adposition; ADV, adjective; CONJ, conjunction; PRON, pronoun.

misleading information. Second, in a real-world context the number of preceding and following tags that are missing might be higher.

Below we are going to describe BN and RF, the latter being used as a benchmark.

3.1 | The Bayesian network

The BN is a model that uses a directed acyclic graphs (DAG) to make explicit a set of (conditional) dependence and independence properties among the variables represented in the BN. BNs have been widely used in different domains such as biology [35], consumer satisfaction [36], product perception [37], and so on. A DAG $\mathcal{G}$ is formed by the pair $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a finite set of distinct vertices, $\mathbf{V} = \{V_i\}_{i=1}^{k}$, which correspond to a set of random variables $\mathcal{X} = \left\{X_{V_i}\right\}_{i=1}^{k}$ indexed by $\mathbf{V}$, and $E \subseteq \mathbf{V} \times \mathbf{V}$ is the set of directed edges between pairs of nodes in $\mathbf{V}$. For any couple of variables $V_i, V_j \in E$ linked by an arrow ($\rightarrow$) from node V_i to node V_j, V_i is said to be a parent of V_j whereas V_j a child of V_i. The set of parents of a node V is denoted by $pa(V)$. A BN over the variables in $\mathcal{X}$ is defined as the triplet $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P})$, where $\mathcal{G}$ is a DAG and $\mathcal{P}$ is a set of conditional probability distributions containing one distribution of the $P\left(X_v|X_{pa(v)}\right)$ kind for each $X_v \in \mathcal{X}$, where $X_{pa(v)}$ is the set of parent variables of X_v. The joint probability distribution of $\mathcal{X}$, $P(\mathcal{X})$, is factorized as follows:

$$P(\mathcal{X}) = \prod_{v\in V} P\left(X_v|X_{pa(v)}\right). \tag{1}$$

A BN can be seen in terms of its DAG, the qualitative component, and of its joint probability distribution (1), the quantitative component. The arrows direction in the DAG does not necessary have a causal interpretation, but it shows the way in which the information can flow into the BN.

The construction of a BN consists in identifying the DAG and specifying the joint probability distribution in terms of the set of conditional probability distributions $P\left(X_v|X_{pa(v)}\right)$. The DAG can be derived manually, that is be elicited by experts of the field, or be automatically derived from data. To automatically find a BN's structure, one can follow constraint-based, score-based, or hybrid algorithms. Constraint-based algorithms use conditional independence tests, focusing on the presence of individual arcs; score-based algorithms assign scores to evaluate DAGs as a whole; hybrid algorithms combine constraint-based and score-based algorithms. In this study, we considered two score-based algorithms, the Hill Climbing (HC) and the Tabu search (TABU), which belong to the class of greedy search algorithms. Both algorithms explore the search space starting from an empty DAG and adding, deleting, or reversing one arc at a time until the score being examined can no longer be improved [38]. TABU depends on a tabu list containing a number of tabus that prevent revisiting recently seen structures when going away from local optima. We took into account two different scores: the Bayesian Information criterion (BIC) and the Bayesian Dirichlet equivalent uniform score, or BDE [39]. The latter depends on a parameter called imaginary sample size (iss) associated with a Dirichlet prior, which defines how much weight is assigned to the prior in the form of the size of an imaginary sample supporting it.

A BN can be used to answer questions related to the domain of the data; this process is also known as probabilistic reasoning or belief updating. It focuses on the calculus of posterior probabilities given a new piece of information called evidence $\mathbf{E}$. The kind of evidence considered in this paper is called hard evidence, that is, an instantiation of a subset of or all the variables in the BN: $\mathbf{E} = \left\{X_{i_1} = e_1, \ldots, X_{i_m} = e_m\right\}$, with $i_1 \neq \cdots \neq i_m \in \{V_1, V_2, \ldots, V_k\}$. Hard evidence can come from new partial or complete observations recorded after the BN was learned. Therefore, from a learned BN, it is possible to compute the effect of $\mathbf{E}$ on one or more target variables

$\mathbf{X}'$ using the knowledge encoded in the BN, by computing the posterior distribution $P(\mathbf{X}'|\mathbf{E})$. In this article, we used the junction tree representation of the BN $\mathcal{N}$, where a junction tree is a transformation of the moral graph of $\mathcal{G}$ (the moral graph of a DAG is an undirected graph built adding edges between all pairs of non-adjacent nodes that have a common child and ignoring the directions of all the edges in the graph) in which the nodes were clustered to reduce any network structure into the tree [38]. The belief updates were performed efficiently using Kim and Pearl's Message Passing algorithm which involves the repeated application of Bayes' Theorem and the use of conditional independencies encoded in the network structure [40].

So, an interesting aspect in the use of BN to compute posterior probabilities given a new piece of information $\mathbf{E}$, is that $\mathbf{E}$ can concern any subset of the available variables. For example, if the variables are the ones in $Tag_{t-/+3}$ and one wants to predict the posterior probability $P(tag_t|\mathbf{E})$, $\mathbf{E}$ can be a single variable in $Tag_{t-/+3}$, a couple, such as (tag_{t-1}, tag_{t+1}) or (tag_{t-3}, tag_{t+3}), a triple, such as $(tag_{t-2}, tag_{t-1}, tag_{t+1})$, or the entire set $Tag_{t-/+3} - tag_t$. This makes the BN a suitable tool for determining tag_t when surrounding tags are missing, without the need of a preliminary imputation step for these missing tags.

Starting from the set of attribute probabilities of the target variable estimated by the BN, it is necessary to synthesize it in order to extract a predicted attribute. In this article, the target variable is the unknown POS tag tag_t, which is a nominal variable, and the used classifiers are:

- *Bayes classifier* (BC): It assigns a word to the most likely POS tag.
- *Maximum difference classifier* (MDC): It evaluates the difference between the predicted probabilities of each possible attribute of the POS tag variable and the corresponding sample frequencies derived from the dataset. It takes the attribute corresponding to the maximum difference among all attributes [41, 42].
- *Hybrid classifier* (Hybrid): It is created to exploit the benefits of either the BC or MDC; it uses MDC when each frequency associated with the modal attribute is less than 0.5; otherwise it uses BC.

3.2 | Random forest

RF is a machine learning algorithm proposed by Reference [9] and belonging to the family of ensemble learning models with a decision tree [43] as base learner. Given that the target variable in the POS tag context is nominal, the appropriate decision tree is a classification tree in which the data space generated by p explanatory variables X_i is recursively partitioned into a set of regions that are homogeneous with respect to a target variable, and assigning each region, called a leaf node, to the class occurring more often. RF consists in drawing B bootstrap samples from the data and growing a tree on each one with the following perturbation: before each split, one randomly selects $m \leq p$ explanatory variables from the p available as candidates for splitting. Then, for each tree the predicted class is recorded, and the majority vote is taken as the overall RF prediction. Three parameters need to be set: the number B of bootstrap samples, the number k of variables to be selected at each split, and the minimum number of observations presented into a leaf node. In general, a high B value does not cause overfitting problems, whereas the default value for m is the integer part of k, and the default minimum node size is one.

RF tends to achieve good classification results even when using its default parameters [44]. In a recent and large comparison study, the RF classifier was ranked as the best classifier among 17 main machine learning algorithms [10], and that is why we chose it as benchmark for the analysis of full information.

RF handles missing values by applying either the built-in methodology of the surrogate splits or a preliminary imputation step; a comparison of the two approaches can be found in [45]. Reference [46] describes three strategies for missing data imputation in which missing data are pre-imputed or simultaneously imputed, also when the forest is used for prediction. It is important to notice that both approaches imply a “fill in” step for missing data, unlike BN.

3.3 | Performance metrics

When the number of attributes of the target variable is greater than two and in case of unbalanced classes, as in this context, considering several metrics is better for the evaluation of the predictive performance of a model. In fact, the simple percentage of correct predictions gives a limited view of its predicting abilities. Therefore, to give a synthetic measure of the algorithm predictive capabilities we adopted, following [47], the metrics: area under the curve (AUC) of the receiver operating characteristic (ROC) curve, average accuracy, macro precision, macro-averaging F1-score and overall accuracy.

The AUC is based on the ROC curve, which plots the false positive rate in x-axis, versus the true positive rate in the y-axis. The AUC $= \int \mathrm{ROC}dT$ measures the global discriminatory performance of a classifier. ROC curves can be applied to unbalanced tasks and without knowing a priori the false positive and false negative costs [48]. The AUC metric was first studied for the binary classification task, but later it showed its potential also for the multiclass classification [49, 50]. In this work, following the research of [50], the AUC reported is the average value obtained on single pairwise ROC curve for each of the 10-fold experiments for the prediction of the unknown tag_t.

In a multiclass classification problem with $l > 2$ attributes, the $l \times l$ confusion matrix can be reduced to l 2×2 confusion matrices, one for each class label $i = 1, 2, \ldots, l$, from which accuracy, precision, recall (or sensibility), and F1-score can be calculated as follows:

$$Acc_i = \frac{tp_i + tn_i}{tp_i + tn_i + fn_i + fp_i}$$

$$Prec_i = \frac{tp_i}{(tp_i + fp_i)}$$

$$Recall_i = \frac{tp_i}{(tp_i + fn_i)}$$

$$F1 - score_i = 2 \times \frac{\text{Prec}_i \times \text{Recall}_i}{\text{Prec}_i + \text{Recall}_i},$$

where tp_i, fp_i, fn_i, and tn_i are the number of true positives, false positives, false negatives, and true negatives for the ith class. The average accuracy (Av Acc) computes the average per-class effectiveness of the classifier, the macro precision (M Prec) denotes the average per-class precision whereas the macro average F1 score (MAF1) [51] is the average per-class F1 scores. The F1-score is considered a more reliable measure when data are unbalanced, as in our case. These metrics were obtained averaging the l corresponding indicators as follows:

$$AvAcc = \frac{\sum_{i=1}^{l} Acc_i}{l}$$

$$MPrec = \frac{\sum_{i=1}^{l} Prec_i}{l}$$

$$MAF1 = \frac{\sum_{i=1}^{l} F1 - score_i}{l}$$

Moreover, we also reported the overall accuracy (Acc) for all classes, that is, the simple sum of the correct classified, or predicted, cases over the total number of cases n:

$$Acc = \frac{\sum_{i=1}^{l} tp_i}{n}.$$

3.4 | The auxiliary POS tagger

To identify the proposed auxiliary POS tagger, we used the *Brown Corpus*, a wide and well-known Corpus compiled in the 1960s at Brown University. It is a general corpus (text collection) in the field of corpus linguistics containing 500 English-language text samples, totaling roughly one million words. The tagged Brown Corpus used a selection of about 80 POS tags, as well as special indicators for compound forms, contractions, foreign words, and a few other phenomena. In this analysis we used the version available in the *nltk* module of *Python* [52] composed of 1.15 million words tagged by 12 basic POS tags. This tagset is called Universal POS tagset and was introduced by Reference [53]. Before implementing the BN, we customized the notation for punctuation, since in the Brown universal tag all punctuation is defined as "." In order to differentiate strong punctuation symbols identifying the end of a grammatical sentence, we kept the tag "PUNCT" for the following punctuation: {. : ! ?}, whereas we created a tag "COMMA" for the "," Moreover, we removed some punctuation symbols that did not bring useful information for the tag sequence, such as: {"< () " " : - – %}. The class represented by "X" refers to others POS tags as foreign words. Table 3 reports the tagset on which the auxiliary POS tagger is based and the respective meaning.

TABLE 3 Tagset for the auxiliary Part-of-Speech (POS) tagger

POS	Brown
Adjective	ADJ
Adposition	ADP
Adverb	ADV
Conjunction	CONJ
Determiner	DET
Noun	NOUN
Number	NUM
Pronoun	PRON
Particle	PRT
Verb	VERB
Other	X
,	COMMA
. , : , ! , ?	PUNCT

To determine the POS tag of the unknown token, the collection of POS tags reported in Table 3 was used as an informative tags sequence. Moreover, since the application of basic regular expression modules allows identifying digits and punctuation marks, we did not estimate the unknown tag_t whenever the token was NUM, COMMA, or PUNCT.

Table 4 reports the marginal frequencies in the Brown Corpus for the set of unknown tag_t, showing a highly unbalanced POS tags distribution; the mode corresponds

TABLE 4 Observed POS tag frequencies for tag_t in Brown Corpus

POS tag	Brown
ADJ	0.084
ADP	0.145
ADV	0.056
CONJ	0.038
DET	0.137
NOUN	0.276
PRON	0.049
PRT	0.030
VERB	0.183
X	0.001

Abbreviations: ADJ, Adjective; ADP, adposition; ADV, adjective; CONJ, conjunction; DET, determiner; PRON, pronoun; PRT, particle.

to NOUN, even if the frequency is below 50%, followed by VERB.

The identification of the suitable BN was conducted in two steps. First, we identified the best DAG using the three different sets of preceding/following tags, then we performed a 10-fold cross-validation procedure to define the auxiliary POS tagger. Operationally, we used three R packages: *bnlearn* [54] for BN's identification and estimation, and *rminer* [55] and *pROC* [56] for predictive performance evaluation.

3.4.1 | First step

In order to identify the structure of the relations between the POS tags involved, we used the HC and TABU algorithms, and the BIC and BDE scores. Since the TABU algorithm and the BDE score depend on the tabu and iss parameters, respectively, we considered a set of possible values for these two parameters: {5, 10, 20, 50, 100, 200, 300, 500, 700, 1000} for tabu and {500,1000,2500,5000,10000,20000} for iss. Figure 1 reports the best DAG structures for each of the sets $Tag_{t-/+1}$, $Tag_{t-/+2}$, and $Tag_{t-/+3}$ originated from a 10-fold cross-validation step.

In this phase, we applied the BC to predict the POS tag. Since the dataset includes about 1 million tokens, the training set at each iteration is of about 900,000 tokens and the test of 100,000 tokens. When the information considered only one previous/next tag, the best BN structure was provided using the HC algorithm with the BDE score (iss = 500); if there were two preceding/-following tags, the best model was obtained using the HC algorithm with the BIC score, whereas in case there were three preceding/following tags, the best BN structure was provided using the HC algorithm with the BDE score (iss = 5000).

For all three DAGs shown in Figure 1, the set of arcs well represents the expected relationships among the different POS tags. The estimated DAG structure for $Tag_{t-/+1}$ (Figure 1A is fully connected. Looking at Figure 1C, there is an arrow that can seem anomalous: **Tag t → Tag t−1**, nevertheless, the BN derived in this paper is not a causal BN, therefore, the arrows direction in the DAG does not have a causal interpretation, but shows, as previously mentioned, the way in which the information can flow into the BN.

3.4.2 | Second step

To decide which combination of information set, BN, and predictive criterion to use, we performed a 10-fold cross-validation procedure, following the procedure proposed by [57]. For each fold, we reckoned the three previously described predictive criteria and the evaluation metrics recalled in Section 3.1. Table 5 shows the mean and standard deviation (in parenthesis) of the obtained values.

Through the analysis of the reported results we identified, as the overall best model, the BN built using the information set $Tag_{t-/+3}$, the DAG of Figure 1C, and MDC. This combination has, on average, the best predictive performance, according to the considered metrics.

4 | EXPERIMENTS

In order to test the validity of the proposed method in application contexts, we performed several analyses. In the first one, we analyzed the Brown Corpus as well as a more modern dataset called ARK. We applied to both datasets the proposed auxiliary POS tagger and the RF to solve the problem of predicting an unknown tag, knowing the entire sequence of three preceding and following tags. We considered the RF as a benchmark for the proposed POS tagger based on BN and the analyses were carried out using the *sklearn* module on *Python*. The second analysis was inspired by the real-world example shown in Table 2, in which the entire sequence of three preceding and following tags was not always available. Therefore, we evaluated the predictive performance of the proposed auxiliary POS tagger in the case of missing tags. In the last analysis, we applied the proposed method to a small corpus of Web 2.0 data, which included Twitter messages, Facebook posts, and Tripadvisor and Amazon reviews.

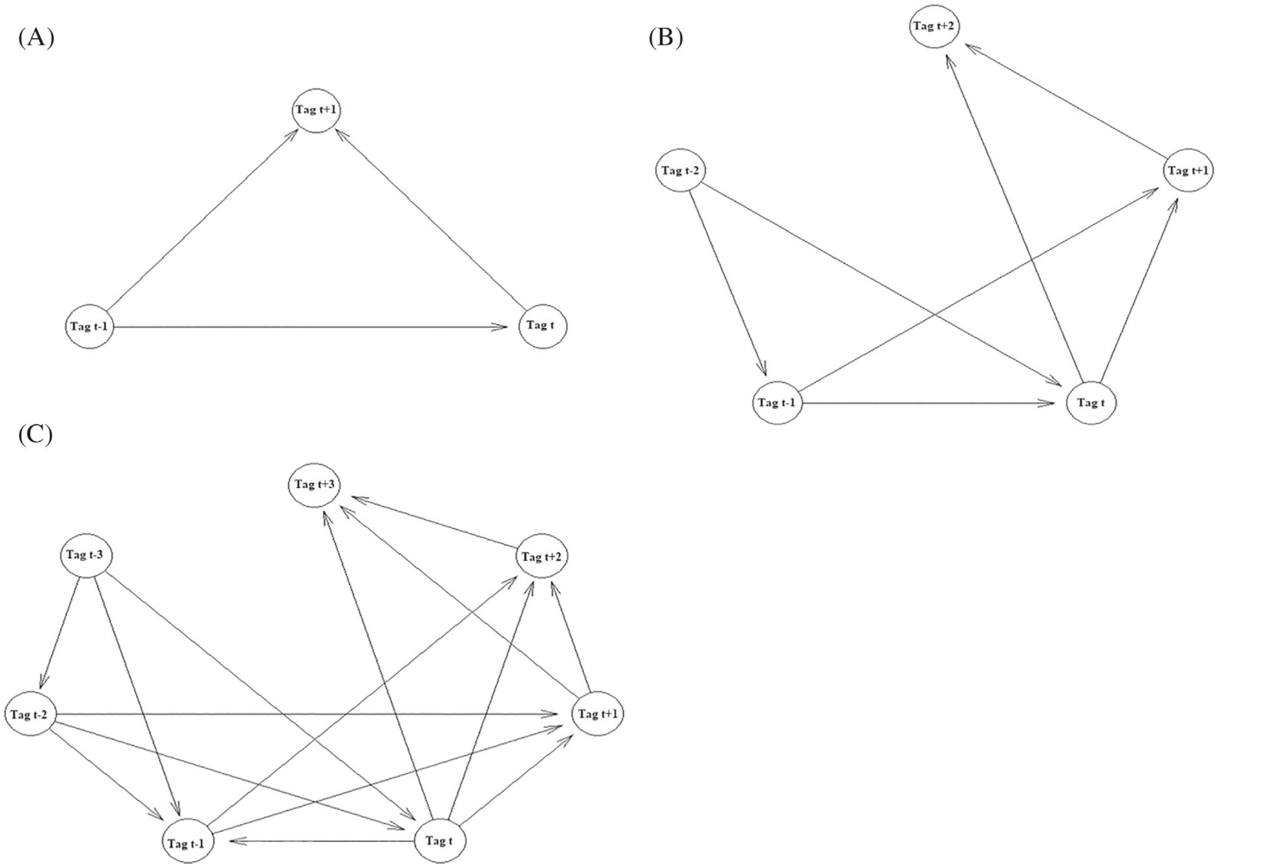

FIGURE 1 DAG structure for the information sets (A) $Tag_{t-/+1}$, (B) $Tag_{t-/+2}$, and (C) $Tag_{t-/+3}$

TABLE 5 Mean (SD) of the best BN models for the different tag sequences

Information set	Method	AUC	Av Acc	MAF1	M Prec	Acc
$Tag_{t-/+1}$	BC	0.728 (0.020)	0.410 (0.007)	0.470 (0.017)	0.310 (0.007)	0.572 (0.015)
	MDC	0.728 (0.020)	0.410 (0.007)	0.470 (0.017)	0.310 (0.007)	0.572 (0.015)
	Hybrid	0.728 (0.020)	0.410 (0.007)	0.470 (0.017)	0.310 (0.007)	0.572 (0.015)
$Tag_{t-/+2}$	BC	0.720 (0.016)	0.413 (0.007)	0.500 (0.018)	0.313 (0.007)	0.613 (0.015)
	MDC	0.732 (0.015)	0.435 (0.007)	0.515 (0.018)	0.335 (0.007)	0.606 (0.016)
	Hybrid	0.730 (0.015)	0.434 (0.007)	0.514 (0.018)	0.334 (0.007)	0.607 (0.016)
$Tag_{t-/+3}$	BC	0.717 (0.015)	0.426 (0.006)	0.523 (0.015)	0.326 (0.006)	0.630 (0.013)
	MDC	0.731 (0.015)	0.444 (0.007)	0.533 (0.017)	0.344 (0.007)	0.624 (0.015)
	Hybrid	0.730 (0.015)	0.443 (0.007)	0.533 (0.017)	0.343 (0.007)	0.625 (0.015)

Abbreviations: Acc, overall accuracy; AUC, area under the curve; Av Acc, average accuracy; BC, Bayes classifier; MAF1, macro average F1 score; MDC, maximum difference classifier; M Prec, macro precision.

4.1 | Full data: Brown and ARK datasets

The *ARK dataset* includes 39 K tokens from Twitter messages. It was developed by [58] and originally included 20 POS tags plus 5 specific POS tags linked to Twitter writing. In our analysis, we focused on a partition of ARK called "Daily547" which included 547 tweets, one per day from January 2011 through June 2012.[2] Since the tagset used in ARK is larger than the one in the Brown Corpus, we aggregated some POS tags to comply with the Brown notation. Table 6 reports the aggregations

[2]https://github.com/brendano/ark-tweet-nlp/tree/master/data/twpos-data-v0.3

TABLE 6 Part-of-Speech (POS) tag levels (tagset) in Brown Corpus and ARK dataset

POS	Brown	ARK
Adjective	ADJ	A
Adposition	ADP	P
Adverb	ADV	R
Conjunction	CONJ	&
Determiner	DET	D
Noun	NOUN	N, ^, Z, S, M, #, @
Number	NUM	$, PM
Pronoun	PRON	O
Particle	PRT	T, X,Y, L
Verb	VERB	V
Other	X	U, G, !
,	COMMA	–
. , ; , ! , ?	PUNCT	~, E

performed. The specific Twitter tags (#, U, E, ~, @) were rearranged as follows: hashtag and mentions (#, @) were recorded as nouns, emoticons (E) and discourse markers (~) as punctuation, and URLs (U) as "Other." The POS "Other" included interjections (!) and foreign words, symbols, and other words (G). Marginal frequencies of the POS tags in the ARK dataset were similar to the ones of the Brown Corpus, with the exception of the DET tag (Brown: 0.137, ARK: 0.074) and the X tag (Brown: 0.001, ARK: 0.064). These differences highlight the different communication styles of a classical text (Brown) and a microblog text (ARK).

Starting from the DAG identified in Figure 1C, we estimated the corresponding BN on the ARK dataset. Then, we trained the RF on Brown Corpus and ARK dataset keeping the same information set $Tag_{t-/+3}$. We adopted the procedure proposed by [57], that is we firstly calculated the metrics (Av Acc, AUC, M Prec, MAF1, Acc) for each fold of a 10-fold cross-validation procedure and then reckoned the average of the 10 different results. The two algorithms (BN and RF) were evaluated both for the Brown Corpus and the ARK dataset. Moreover, we investigated their performance in a domain adaptation structure, where the training domain was the Brown Corpus and the target domain the ARK dataset. This domain adaptation analysis is of particular interest because there currently are only few and relatively small labeled datasets to train the models for Twitter and Web 2.0 data. Table 7 reports the results in terms of evaluation metrics for the Brown, ARK, and domain adaptation cases, comparing the BN and RF models.

TABLE 7 BN and RF comparison for Brown, ARK and domain adaptation

	Model	AUC	Av Acc	MAF1	M Prec	Acc
Brown → Brown	BN	0.731	0.444	0.533	0.344	0.624
	RF	0.717	0.421	0.520	0.321	0.628
ARK → ARK	BN	0.629	0.355	0.364	0.259	0.474
	RF	0.615	0.324	0.300	0.234	0.492
Brown→ ARK	BN	0.613	0.337	0.318	0.242	0.399
	RF	0.614	0.304	0.308	0.217	0.452

Abbreviations: Acc, overall accuracy; AUC, area under the curve; Av Acc, average accuracy; MAF1, macro average F1 score; M Prec, macro precision.

The results for the Brown Corpus are overall better than the ones for the ARK dataset, both for BN and RF. Moreover, when comparing the performance of the models that used only the ARK dataset (ARK → ARK) with respect to the cross-domain setting (Brown → ARK), we noticed only a slight decrease in the evaluation metrics. For example, the AUC of BN for ARK → ARK was only 2.6% points greater. This outcome shows that the use of models estimated on a wide corpus, such as the Brown Corpus, to predict unknown POS tags for Web 2.0 datasets, does not entail a significant reduction in the predictive power of the models; therefore, they can be successfully applied to datasets other than the Brown Corpus. As for the comparison of the BN and RF performance, we observed that all metrics have comparable values, indicating a similarity in the performance. These findings confirmed the robustness of the proposed method. The auxiliary POS tagger has the advantage that it can be used even when there are missing tags without an imputation step, which is needed if RF is used. The latter is further explored in the section below.

4.2 | Data with missing tags

Taking inspiration from the real-world example shown in Table 2, we tested the model prediction capabilities when some preceding/following tags are missing. To this end, we performed a 10-fold cross-validation procedure described below. For each fold, the training set was used to estimate the BN corresponding to the DAG of Figure 1C. The "unknown" tags (i.e., tag_t in the set $Tag_{t-/+3}$) in the test set were predicted under the hypothesis that the tags listed in Table 8 were missing, and applying MDC. Then, we computed the metrics Av Acc, AUC, M Prec, MAF1, and Acc; Table 8 reports the average values of the metrics over the 10 distinct results, with standard deviation in parenthesis.

TABLE 8 Mean (SD) of the best BN model with missing tags

Missing tags	AUC	Av Acc	MAF1	M Prec	Acc
$Tag_{t-3}, tag_{t-2}, tag_{t+2}, tag_{t+3}$	0.728 (0.020)	0.410 (0.007)	0.470 (0.018)	0.310 (0.007)	0.572 (0.015)
$Tag_{t-3}, tag_{t-2}, tag_{t-1}$	0.703 (0.015)	0.352 (0.007)	0.344 (0.015)	0.252 (0.007)	0.442 (0.013)
$Tag_{t+1}, tag_{t+2}, tag_{t+3}$	0.627 (0.015)	0.332 (0.007)	0.323 (0.017)	0.232 (0.007)	0.444 (0.011)
$Tag_{t-2}, tag_{t-1}, tag_{t+1}$	0.601 (0.017)	0.263 (0.010)	0.197 (0.016)	0.163 (0.010)	0.283 (0.014)
$Tag_{t-1}, tag_{t+1}, tag_{t+2}$	0.566 (0.013)	0.256 (0.009)	0.191 (0.015)	0.156 (0.009)	0.290 (0.009)
Tag_{t-1}, tag_{t+1}	0.607 (0.016)	0.292 (0.009)	0.242 (0.017)	0.192 (0.009)	0.330 (0.012)
Tag_{t-3}, tag_{t+3}	0.732 (0.013)	0.440 (0.007)	0.522 (0.020)	0.340 (0.007)	0.612 (0.017)

Abbreviations: Acc, overall accuracy; AUC, area under the curve; Av Acc, average accuracy; MAF1, macro average F1 score; M Prec, macro precision.

The worst performance occurred when both tag_{t-1} and tag_{t+1} were missing, with a 20% reduction of the AUC and 50% reduction of the other indicators. There was a negligible reduction in the metrics when both tag_{t-3} and tag_{t+3} were missing, whereas the reduction was limited when tag_{t-1} or tag_{t+1} was missing.

4.3 | Web 2.0 data

This section reports the results of a practical application of the proposed auxiliary POS tagger to new data. Given a corpus of manually labeled Web 2.0 data, first we tagged it with an online dictionary and then predicted the missing POS tags applying the BN defined in previous sections. We then compared the results of the proposed method with three other POS taggers: the Stanford Log-linear, the TreeTagger, and the TextBlob Python library [59]. The tagging step performed by the TreeTagger left some tokens unlabelled, so the proposed auxiliary POS tagger was applied also in those cases.

The Web 2.0 dataset includes messages from Twitter, Facebook, Tripadvisor, and Amazon and aims to cover the most used platforms in Web 2.0 communication. Tripadvisor and Amazon data were collected in January 2018, and Facebook and Twitter data in May 2018. These data are publicly available at https://github.com/sgol17/Web2.0-data-for-POS-tagging. In the first step, the messages were tagged using the online Cambridge dictionary.[3] The POS tagset used by the online Cambridge Dictionary is similar to the Brown one, with only a few differences regarding

1. modal verbs, which were affiliated to VERB,
2. plural nouns, which were affiliated to NOUN,
3. exclamations, which regard words such as “ok”, were attributed to the POS more pertinent to each particular context.

[3] https://dictionary.cambridge.org/dictionary/english/

All POS taggers (i.e., Stanford Log-linear, TreeTagger, and TextBlob) tagsets were adapted according to the one in Table 6, in order to homogenize the experiments.

Table 9 reports the overall accuracy and the AUC of the proposed method and of the other three POS taggers applied to the data examined. Looking at the results on the data from the two websites and Facebook, the proposed approach (Cambridge Dict. + BN and TreeTagger + BN) outperformed the Stanford Log-linear, TextBlob and TreeTagger in terms of both Acc and AUC, whereas it had a poorer performance on the Twitter dataset in terms of Acc.

Considering the performance of the auxiliary POS Tagger taken alone, we must first underline that the scenario in which the corpus described in this subsection is placed is the one concerning the cross domain and, in a few cases, the one concerning incomplete information. Looking at the Tripadvisor dataset, which is the largest dataset (1914 tags), the values of Acc and AUC were respectively 0.418 and 0.709, when the tagging step was performed by the Cambridge Dictionary, and 0.429 and 0.737 when the tagging step was performed by the TreeTagger. These values are in line with what is shown in Table 7. The same happened with the Amazon dataset (466 tags) for which the Acc was 0.375, whereas for the Facebook dataset (866 tags) the overall accuracy was a bit lower at 0.290. The Twitter dataset, which is the smallest dataset (254 tags), is the one in which the auxiliary POS Tagger performed worse in terms of overall accuracy. The percentage of unlabelled tokens, but correctly labeled by the auxiliary POS Tagger (Acc) was equal to 21.9%, far from what expected. An explanation of this behavior relies on the quality of the preliminary tagging step. In fact, for this dataset, the error rate of the Dictionary was high, equal to the 31.5%, and this can affect the ability to correctly tag the missing tags when some wrong tags are in the set of the tags used by the BN, as happened for some missing tags in this dataset.

With reference to the example of cyber-slang data reported in Table 2, Table 10 shows how the proposed method filled the ? and × symbols. The last column of

TABLE 9 Overall Accuracy and AUC (Acc-AUC) for the Web 2.0 data

	Websites		Social media	
Method	**Amazon**	**Tripadvisor**	**Facebook**	**Twitter**
Cambridge dictionary + BN	0.867–0.943	0.874–0.914	0.847–0.928	0.714–0.915
Stanford log-linear	0.815–0.872	0.809–0.897	0.834–0.906	0.750–0.874
TreeTagger	0.839–0.944	0.858–0.922	0.849–0.915	0.761–0.905
TreeTagger + BN	—	0.862–0.919	0.851–0.915	—
TextBlob	0.821–0.906	0.848–0.891	0.836–0.896	0.775–0.885

Abbreviations: Acc, overall accuracy; AUC, area under the curve; BN, Bayesian network.

TABLE 10 Application of the auxiliary POS tagger to the Twitter cyber-slang data of Table 2

Token	Cambridge dictionary + auxiliary POS tagger	Stanford POS tagger + auxiliary POS tagger	Ground truth POS tags
@Iselgomezl	*PRON*	*PRON*	NOUN
Dont	*VERB*	*VERB*	VERB
Really	ADV	ADV	ADV
Know	VERB	VERB	VERB
Where	ADV	ADV	ADV
I	PRON	*PRON*	PRON
Would	VERB	VERB	VERB
Be	VERB	VERB	VERB
Without	ADP	ADP	ADP
Youuuuuuuuuu	*NOUN*	NOUN	PRON
And	CONJ	CONJ	CONJ
Demi	NOUN	*NOUN*	NOUN
<3 (♥)	*VERB*	*NOUN*	PUNCT
I	PRON	*ADP*	PRON
Luv	NOUN	*ADP*	VERB
Youu	*VERB*	*DET*	PRON
#IfMyMoms HadATwitter	*PRON*	NOUN	NOUN

Abbreviations: ADP, adposition; ADV, adjective; CONJ, conjunction; DET, determiner; PRON, pronoun.

the table reports the ground truth POS tags. The words in italics are the imputed ones. The proposed auxiliary POS tagger, jointly with the Cambridge dictionary, achieved a 64.7% accuracy (11 correct matches out of 17 tokens). The Stanford Log-linear was able to correctly identify 47.1% of the tokens, and this percentage increased up to 64.7% when the auxiliary POS tagger was applied to the tokens with a ×. Analyzing the wrong POS tags assigned by the auxiliary POS tagger, we could notice frequent miss-matches between nouns and pronouns. In fact, without the usage of token information, the role of a noun and pronoun in a sentence is similar: both can be either subject or direct object. Nevertheless, this kind of error came up also when traditional POS taggers were used, as shown by the tag assigned to the word "youuuuuuuuuu" by the Stanford Log-linear. As argued by [60], an error analysis regarding POS tag estimation will help in better understanding the method performance, since not all errors are the same. Moreover, the auxiliary POS tagger abilities might also be affected by the misclassified tags deriving from the online dictionaries or the other POS taggers used in the preliminary labeling phase.

5 | CONCLUSIONS

In this paper, we are proposing an auxiliary POS tagger that aims to predict unknown POS tags given a partially labeled text. Our motivations were mainly related to the inability of traditional POS taggers to correctly assign a POS tag to modern text from blogs and microblogs, characterized by the presence of the so-called cyber-slang. Moreover, the availability of large and labeled Web 2.0 corpora to train the models is still limited. For these reasons, the proposed auxiliary POS tagger was developed avoiding token lexical information, thus aiming to be a flexible tool applicable to different domains.

The proposed auxiliary POS tagger applies BNs to predict the unknown tag_t by analyzing a set of three preceding and following POS tags and using, as a training set, the well-known Brown Corpus. To identify the optimal BN structure, several experiments were conducted, and the structure of the best BN was identified through the score-based algorithm HC. Moreover, to translate the BN probabilities to a POS tag class, we performed a 10-fold cross-validation of three possible classifiers, finding that MDC is the best one.

To test the capabilities of the proposed auxiliary POS tagger, we performed several experiments. We analyzed the ARK dataset and a sample of manually labeled data from Facebook, Twitter, Tripadvisor, and Amazon. The domain adaptation analysis involving the ARK dataset showed a good predictive performance of the model trained on a traditional dataset (Brown Corpus) to estimate the POS tag of modern texts (ARK dataset). The proposed method, jointly with the Cambridge online dictionary, performed well when applied to Web 2.0 data, also compared to three traditional POS taggers (Stanford Log-linear, TreeTagger and TextBlob).

Furthermore, we also tested the method with missing tags (apart from the target tag_t) in the tags sequence, and we observed that the predictive performance considerably decreased only if both tag_{t-1} and tag_{t+1} were missing.

The main advantage of the proposed method is its flexibility toward different domains and kinds of texts, due to its ability to overcome the need of token knowledge. It can be applied to any language, after a preliminary estimation of the BN trained on a large tagged corpus. However, since it is an auxiliary tool, there is the need of a pre-tagging step that can be done through online dictionaries or other POS taggers, and the validity of this preliminary tagging phase has an impact on the performance of the proposed auxiliary POS tagger. Therefore, in future works we aim to solve this limitation in order to create an integrated system able to assign a meaningful POS tag to each word contained into Web 2.0 modern texts, testing other dictionaries and POS taggers.

ACKNOWLEDGMENT

The authors wish to thank Marco Sandri, the two anonymous reviewers and the Associate Editor for their constructive and valuable comments that have significantly enhanced the quality of the paper. Open Access Funding provided by Universita degli Studi di Brescia within the CRUI-CARE Agreement.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available at: Brown Corpus: https://www.nltk.org/nltk_data/; Daily547 (Partition of ARK): https://github.com/brendano/ark-tweet-nlp/tree/master/data/twpos-data-v0.3; Web2.0 data: https://github.com/sgol17/Web2.0-data-for-POS-tagging.

ORCID

Silvia Golia https://orcid.org/0000-0003-0015-8126

REFERENCES

1. Z. Toosinezhad, M. Mohamadpoor, and H. T. Malazi, *Dynamic windowing mechanism to combine sentiment and n-gram analysis in detecting events from social media*, Knowl. Inf. Syst. 60 (2019), 1–18.
2. P. Zola, P. Cortez, C. Ragno, and E. Brentari, *Social media cross-source and cross-domain sentiment classification*, Int. J. Inf. Technol. Decis. Mak. 18 (2019), no. 05, 1469–1499.
3. L. Rocca, D. Giacomini, and P. Zola, *Environmental disclosure and sentiment analysis: State of the art and opportunities for public-sector organisations*, Meditari Accountancy Research, 2020.
4. P. Zola, P. Cortez, and M. Tesconi, "*Using google trends, gaussian mixture models and dbscan for the estimation of twitter user home location*," *International conference on computational science and its applications*, Springer, 2020, pp. 526–534.
5. P. Nand and R. Perera, *An evaluation of pos tagging for tweets using hmm modeling*, Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015) (2015), pp. 83–89.
6. S. Meftah and N. Semmar, *A neural network model for part-of-speech tagging of social media texts*, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018), pp. 2821–2828.
7. I. Alkhairy, S. Low-Choy, J. Murray, J. Wang, and A. Pettitt, *Quantifying conditional probability tables in bayesian networks: Bayesian regression for scenario-based encoding of elicited expert assessments on feral pig habitat*, J. Appl. Stat. 47 (2020), no. 10, 1848–1884.
8. U. B. Kjærulff and A. L. Madsen, *Bayesian networks and influence diagrams: A guide to construction and analysis*, Springer, New York, 2013.
9. L. Breiman, *Random forests*, Machine Learning 45 (2001), no. 1, 5–32.
10. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, *Do we need hundreds of classifiers to solve real world classification problems?* J. Mach. Learn. Res. 15 (2014), no. 1, 3133–3181.
11. T. G. Stewart, D. Zeng, and M. C. Wu, *Constructing support vector machines with missing data*, Wiley Interdisciplinary Reviews: Computational Statistics 10 (2018), no. 4, e1430.
12. M. Smieja, Ł. Struski, J. Tabor, B. Zieliński, and P. Spurek, "*Processing of missing data by neural networks*," *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2724–2734.
13. Greene, B. B. and G. M. Rubin, 1971: Automated grammatical tagging of english.
14. J. Cussens, "*Part-of-speech tagging using progol*," *International conference on inductive logic programming*, Springer, 1997, pp. 93–108.
15. Jurafsky, D. and J. H. Martin, 2018: Speech and language processing (draft). *Chapter A: Hidden Markov Models (Draft of* September 11, 2018*). Retrieved March*, **19**, 2019.
16. A. Ratnaparkhi, "*A maximum entropy model for part-of-speech tagging*," *Conference on empirical methods in natural language processing*, 1996.
17. D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "*A practical part-of-speech tagger*," *Proceedings of the third conference on applied natural language processing*, Vol '92, Association for Computational Linguistics, Stroudsburg, PA, USA, ANLC, 1992, pp. 133–140. https://doi.org/10.3115/974499.974523.
18. H. Schmid, "*Part-of-speech tagging with neural networks*," *Proceedings of the 15th conference on computational linguistics-volume 1*, Association for Computational Linguistics, 1994, pp. 172–176.

19. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "*Feature-rich part-of-speech tagging with a cyclic dependency network*," *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.
20. G. Attardi and M. Simi, *Overview of the evalita 2009 part-of-speech tagging task*, Workshop Evalita, Citeseer, 2009.
21. B. Plank, A. Søgaard, and Y. Goldberg, "*Multilingual part-of-speech tagging with bidirectional long short-termmemory models and auxiliary loss*," *Proceedings of the 54th annual meeting of the Association for Computational Linguistics*, Short Papers, Vol 2, Association for Computational Linguistics, 2016, pp. 412–418.
22. M. Yasunaga, J. Kasai, and D. Radev, "*Robust multilingual part-of-speech tagging via adversarial training*," *Proceedings of the 2018 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies*, Long Papers, Vol 1, 2018, pp. 976–986.
23. X. Zhang, Y. Jiang, H. Peng, K. Tu, and D. Goldwasser, "*Semi-supervised structured prediction with neural crf autoencoder*," *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 1701–1711.
24. W. Khan, A. Daud, K. Khan, J. A. Nasir, M. Basheri, N. Aljohani, and F. S. Alotaibi, *Part of speech tagging in urdu: Comparison of machine and deep learning approaches*, IEEE Access 7 (2019), 38918–38936.
25. W. Khan, A. Daud, J. A. Nasir, T. Amjad, S. Arafat, N. Aljohani, and F. S. Alotaibi, *Urdu part of speech tagging using conditional random fields*, Lang. Resour. Eval. 53 (2019), no. 3, 331–362.
26. Ekbal, A., R. Haque, and S. Bandyopadhyay, 2007: *Bengali part of speech tagging using conditional random field. Proceedings of the Seventh International Symposium on Natural Language Processing, SNLP-2007*.
27. N. Ponomareva, P. Rosso, F. Pla, and A. Molina, *Conditional random fields vs. hidden markov models in a biomedical named entity recognition task*, Proc. of Int. Conf. Recent Advances in Natural Language Processing RANLP (2007), 479–483.
28. L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, "*Twitter part-of-speech tagging for all: Overcoming sparse and noisy data*," *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP*, Vol 2013, 2013, pp. 198–206.
29. T. Brants, "*Tnt: A statistical part-of-speech tagger*," *Proceedings of the sixth conference on applied natural language processing*, Association for Computational Linguistics, 2000, pp. 224–231.
30. J. Giménez and L. Marquez, *Fast and accurate part-of-speech tagging: The svm approach revisited*, Recent Adv. Nat. Language Proces III (2004), 153–162.
31. E. Brill, *Transformation-based error-driven learning and natural language processing:A case study in part-of-speech tagging*, Comput. Linguist. 21 (1995), no. 4, 543–565.
32. G. Attardi and M. Simi, "*Character embeddings pos tagger vs hmm tagger for tweets*," *Proceedings CLiC-it 2016 and EVALITA 2016*, 2016.
33. F. Albogamy and A. Ramsay, *Pos tagging for arabic tweets*, Proc Int Conf Recent Adv Nat Lan Proc (2015), 1–8.
34. D. Koller and N. Friedman, *Probabilistic graphical models: Principles and techniques*, MIT press, 2009.
35. P. Berchialla, S. Snidero, A. Stancu, C. Scarinzi, R. Corradetti, and D. Gregori, *Understanding the epidemiology of foreign body injuries in children using a data-driven bayesian network*, J. Appl. Stat. 39 (2012), no. 4, 867–874.
36. S. Salini and R. S. Kenett, *Bayesian networks of customer satisfaction survey data*, J. Appl. Stat. 36 (2009), no. 11, 1177–1189.
37. E. Cene and F. Karaman, *Analysing organic food buyers' perceptions with bayesian networks: A case study in Turkey*, J. Appl. Stat. 42 (2015), no. 7, 1572–1590.
38. M. Scutari and J.-B. Denis, *Bayesian networks: With examples in R*, Chapman and Hall/CRC, 2014.
39. D. Heckerman, D. Geiger, and D. M. Chickering, *Learning bayesian networks: The combination of knowledge and statistical data*, Mach. Learn. 20 (1995), no. 3, 197–243.
40. K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*, CRC Press, 2010.
41. S. Golia and M. Carpita, "*On classifiers to predict soccer match results*," *ASMOD 2018 proceedings of the international conference on advances in statistical modelling of ordinal data*, S. Capecchi, F. D. Iorio, and R. Simone (eds.), FedOAPress, 2018, pp. 125–132.
42. S. Golia and M. Carpita, "*Categorical classifiers in multi-class classification problems*," *CLADAG 2021 book of abstracts and short papers*, G. C. Porzio, C. Rampichini, and C. Bocci (eds.), Firenze University Press, 2021, pp. 288–291.
43. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, Chapman and Hall, London, 1984.
44. H. Trevor, T. Robert, and F. Jerome, *The elements of statistical learning: Data mining, inference, and prediction*, Springer, New York, NY, 2009.
45. A. Hapfelmeier, T. Hothornb, and K. W. Ulm, *Recursive partitioning on incomplete data using surrogate decisions and multiple imputation*, Computational Statistics and Data Analysis 56 (2012), 1552–1565.
46. F. Tang and I. Hemant, *Random forest missing data algorithms*, Statistical Analysis and Data Mining: The ASA Data Science Journal 10 (2017), 363–377.
47. M. Sokolova and G. Lapalme, *A systematic analysis of performance measures for classification tasks*, Inf. Process. Manag. 45 (2009), no. 4, 427–437.
48. T. Fawcett, *An introduction to roc analysis*, Pattern Recogn. Lett. 27 (2006), no. 8, 861–874.
49. A. Fanjul-Hevia and W. González-Manteiga, *A comparative study of methods for testing the equality of two or more roc curves*, Comput. Stat. 33 (2018), no. 1, 357–377.
50. D. J. Hand and R. J. Till, *A simple generalisation of the area under the roc curve for multiple class classification problems*, Mach. Learn. 45 (2001), no. 2, 171–186.
51. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2016.
52. S. Bird, E. Klein, and E. Loper, *Natural language processing with python: Analyzing text with the natural language toolkit*, O'Reilly Media, Inc, 2009.
53. S. Petrov, D. Das, and R. McDonald, *A universal part-of-speech tagset*, Proceedings of the Eighth International Conference on Language Resources and Evaluation, (2012), pp. 2089–2096.
54. M. Scutari, *Learning bayesian networks with the bnlearn R package*, J. Stat. Softw. 35 (2010), no. 3, 1–22.

55. P. Cortez, "*Data mining with neural networks and support vector machines using the r/rminer tool*," *Industrial conference on data mining*, Springer, 2010, pp. 572–583.
56. X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, *Proc: An open-source package for r and s+ to analyze and compare roc curves*, BMC Bioinformatics 12 (2011), no. 1, 77.
57. N. Oliveira, P. Cortez, and N. Areal, *Stock market sentiment lexicon acquisition using microblogging data and statistical measures*, Decis. Support. Syst. 85 (2016), 62–73.
58. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "*Part-of-speech tagging for twitter: Annotation, features, and experiments*," *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human language technologies: Short papers-volume 2*, Association for Computational Linguistics, 2011, pp. 42–47.
59. S. Loria, *textblob documentation*, Release 0 (2018), no. 15, 2.
60. V. Jatav, R. Teja, S. Bharadwaj, and V. Srinivasan, *Improving part-of-speech tagging for nlp pipelines*, arXiv (2017) *preprint arXiv:1708.00241*.

How to cite this article: S. Golia, and P. Zola, *An auxiliary Part-of-Speech tagger for blog and microblog cyber-slang*, Stat. Anal. Data Min.: ASA Data Sci. J. (2022), 1–15. https://doi.org/10.1002/sam.11596