Selecting optimal features for POS-tagging tasks in English

Malak Rassem and Maria Vittoria Ateri
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart
{st179291, st180748}@stud.uni-stuttgart.de

Abstract

This paper introduces a machine learning approach to deal with the task of part-of-speech (POS) tagging in Natural Language Processing. In this work we present a feature selection method for a conditional random field (CRF)-based POS tagger. The results obtained show which features enhance the model performance and which feature combination scores best in terms of F1-score. [1]

1 Introduction

Part of speech (POS) tagging is the process of assigning a part of speech to each word in a text (Jurafsky and Martin, 2000). POS tagging is a relevant NLP application used in machine translation, word sense disambiguation, information retrieval, and more (Chiche and Yitagesu, 2022). For any POS tagging task, an inventory of tags or "tag set" must be chosen. Consequently, the input to a tagging algorithm is a string of words and a specified tag set, while the output is a single best tag for each token (Ratnaparkhi, 1996).

Automatic taggers have been made since the late 1950s, and resolution of ambiguity has been recognized as the most difficult problem in POS tagging, even though issues related to tokenization and lexical analysis have also played a part (Mitkov, 2005). What makes the disambiguation task even harder is the fact that ambiguous words are highly frequent in texts. Particularly ambiguous common words include: "that", "back", "down", "put" and "set" (Jurafsky and Martin, 2000). Here are some examples of different parts of speech for the same token "back":

Earnings growth took a **back/JJ** seat.

A small building in the **back/NN**.

Dave began to **back/VB** toward the door.

[1] You can find our code here

Several approaches to POS tagging exist, namely rule-based methods, stochastic methods, transformation-based methods, and neural networks (Kumawat and Jain, 2015). The detailed description of how these methods work is out of the scope of this paper. Therefore, only a short overview of the stochastic approach is provided in the current discussion.

The stochastic approach entails computation of word and n-gram frequencies and probability distributions. The Hidden Markov Model (HMM) falls into this category, and it has been adopted as baseline in the current project. The Conditional Random Field model (CRF) also falls into this category, and it has been adopted as a more advanced method in the current project. The functioning of both models is described in Section 2.

In this paper, we present a system of POS tagging for English using supervised machine learning. First, a feature analysis is conducted on a sklearn_crfsuite-implemented CRF. This analysis entails a forward selection which provides useful information regarding the effect that each feature has on the final precision, recall, and F1-score, and therefore identifies which features improve the overall performance of the model, and which ones worsen it. Second, a comparison among our HMM, the CRF without features, and the CRF with the selected features confirms the claim (ACLWeb, 2019) that the last one outperforms the other two.

2 Tagging models

In this section we briefly introduce the two tagging models adopted in this work, namely HMM and CRF.

2.1 POS Tagging with HMM

The HMM (Figure 1) is a sequence labeling algorithm, and the aim of a sequence labeler is to assign a label to each unit in a sequence, thus mapping a sequence of observations to a sequence of labels

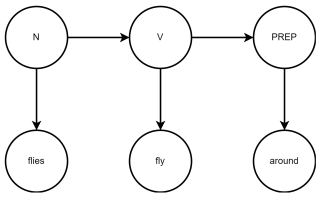

Figure 1: Structure of HMM for POS tagging.

of the same length. Being a probabilistic sequence model, given a sequence of units (words in the case of POS tagging), it computes a probability distribution over possible sequences of labels (the POS tags in the case of POS tagging) and chooses the most probable label sequence. This task is called decoding.

A HMM has two fundamental matrices: one contains the tag transition probabilities P($t_i|t_{i-1}$) which represent the probability of a tag occurring given the previous tag; the other matrix contains the emission probabilities P($w_i|t_i$), which represent the probability, given a tag, that it will be associated with a given word (Jurafsky and Martin, 2000).

The decoding algorithm for HMMs is the Viterbi algorithm, which, given the probability data in the transition and emission matrices, identifies the choice of states that allow the joint probability P(tag,word) to reach the maximum (Jurafsky and Martin, 2000).

HMMs are based on Markov chains. In the current work, a first order Markov chain has been adopted, and therefore the state transition probabilities depend only on the preceding state q_{i-1}. And this is expressed by the first order Markov property (Nilsson, 2005), namely

$$P(q_i|q_1, \ldots, q_{i-1}) = P(q_i|q_{i-1})$$

. In order to handle the problem of underflow, we store the exponential values of the log-probabilities rather than repeatedly taking log. This is supported by Sung-Hyun et al. (2018).

2.2 POS Tagging with CRF

CRFs are a discriminative probabilistic model used to label sequenced data. The main difference between HMMs and CRFs is that the former are generative models, while the latter are discriminative models. This means that CRFs define a conditional probability P(tag|word), instead of a joint probability P(tag,word).

Using CRFs for sequence labeling tasks presents some advantages. One of these advantages consists in the fact that CRFs allow the incorporation of data-dependent global features into the model, which can be hard to do with generative models (Atmakuri et al., 2018).

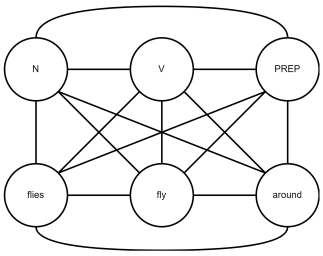

Figure 2: Structure of general CRFs.

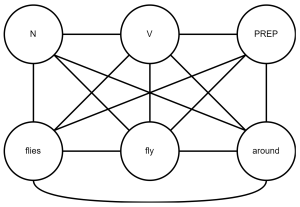

Figure 3: Structure of linear CRFs.

Moreover, general CRFs (Figure 2) do not rely on the assumption of label independence. Nevertheless, the variant of CRF that we use is the linear CRF (Figure 3), in which there are no dependencies between non-consecutive tags. This makes it more computationally- and space-efficient, but is still much more flexible than HMM.

3 Dataset

The tagset used to train the tagger is the Penn Treebank POS Tagset, which has in total 48 tags (see Figure 4). However, we delete three POS tags, namely AFX, XX, and NFP. The reason behind it lies in the fact that these tags appear with very low frequencies (less than 15 occurrences) in the training set, and do not appear at all in the test set.

The dataset used for training contains 731,579 token-POS pairs and the one used for testing contains 33,303 token-POS pairs.

4 Methods

As a POS classification model, we implement a linear-chain (first-order Markov) CRF using the open-source Python library sklearn-crfsuite (Pedregosa et al., 2011). For optimizing the parameters, a Newton-based gradient descent method is used (Pytlak, 2009).

The features used in the CRF models are usually selected by intuition. However, the significance

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Figure 4: The Penn Treebank POS tagsset

-2	-1	0	+1	+2
X_{-2}	X_{-1}	X_0	X_1	X_2

Table 1: Context window. X_0 is the word of interest.

of different features is not always intuitive, which might become problematic as too many features lead to overfitting but too few features might not be sufficient for the task.

We therefore aim to select robust features that can learn the structures present in the data (Stewart et al., 2008). For this purpose, we present our approach to the problem of feature selection.

4.1 Features

To derive the features for the word of interest within a sentence, a context window is introduced. That means, not only the characteristics of the word of interest are considered, but also the ones of the surrounding words, as context has a significant impact on the resulting POS tag. The window chosen here has a width of five words, namely the word, the two previous words and following two words, see Table 1. The feature sets we consider for the word of interest are:

1. the word in lowercase (31,551 individual features);
2. if the word starts with an uppercase letter;
3. if the word ends with -st;
4. if the word ends with -er;
5. if the word ends with -s;
6. the last two letters of the word (712 individual features);
7. if the word is a cardinal number , meaning that it is either a digit or a number written as string (e.g. ‘million’)”;
8. if the word is just a digit;
9. if the word is foreign;
10. if the word is a punctuation mark;
11. the length of the word;
12. if the word is the first word of a sentence (since the training and test sets are lists of sentences);
13. if the word is the last word of a sentence.

With regard to the 9th point, in order to detect foreign words, we define a function that checks whether the word appears paired with the FW tag in the training set. Additionally, the criteria 1, 2, 6, 7, 9 and 10 applied to all the other words in the context window are also taken into the list of features. Each feature set is incrementally added to the feature list, giving us a total of 37 iterations.

Moving on we will refer to feature descriptions as $X_n : F$, where n is the position of the context word and F is the feature number mentioned above.

4.2 Feature Analysis

As stated previously, our goal is to find which feature combination scores best overall on the test set.

The performance is quantified in terms of precision, recall, and F1-score, each for every POS tag. To obtain the overall performance, either the macro average or the weighted average over all POS performances can be taken. If the former is taken, all classes equally contribute to the final averaged metric, whereas if the latter is taken, each class’ contribution to the average is weighted by its size (Khalusova, 2019). The precision P, recall R, and F1 score $F1$ are defined as

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

$$F1 = \frac{2PR}{P + R}.$$

To test the significance of our features, we train our model on the training dataset, and we obtain precision, recall, and F1-score from the test dataset.

Our method consists in starting the training of the model with the first feature set (the word in

lowercase). Consecutively, we iteratively add another feature set in the same order as described in Section 4.1, and retrain to obtain differences in the performance measures. Once all features for the word of interest are taken into account, we also add the feature sets applied to the context words.

More specifically, the filter method used to perform feature selection in our work is called forward selection: in each iteration, we keep the added feature if it improves the result of our model. If an addition of a new feature worsens the performance, the feature is removed in the next iteration (Ray, 2020). In our work, the forward selection is based on the macro F1-score. Nevertheless, we report both the macro F1-score values and the weighted F1-score in order to observe differences due to the imbalanced classes.

The trend of the performance measures provide insight on the relevance of each feature set.

5 Results

Figure 5 and 6 show the evolution of the macro- and weighted-averaged F1-score results, respectively, obtained on the test data. Observe that the F1-score improves significantly after the first few iterations. The most significant increase in the macro F1-Score is at the 11^{th} iteration which corresponds to adding the feature X_0:12.

The maximum macro-averaged F1-score value arises in the 35^{th} iteration, which corresponds to the features X_0:1, X_0:2, X_0:3, X_0:6, X_0:9, X_0:10, X_0:11, X_0:12, X_0:13, X_1:2, X_1:6, X_{-2}:1, X_2:9. In contrast, the maximum weighted-averaged F1-score value arises in the 31^{st} iteration, which corresponds to adding the feature X_2:1. It is noteworthy that this feature decreases the macro F1-score. From the figures we can also see that the weighted-average is much higher than the macro-averaged F1-score. This is due to the fact that the dataset is imbalanced: some classes occur more frequently than others. This is to be expected since this is how English as a natural languages is (Baayen and Lieber, 1996). Moreover, the weighted-average trend is also more stable with less fluctuations in its results. This may mean that individual features added affect smaller classes more.

Furthermore, our experiment shows that eleven POS classes do not exhibit any changes in their F1-score over the iterations. This suggests that such classes, which are typically punctuation and symbols, but also more defined classes such as the "Existential there" class (EX), do not require any features in order to output more accurate results.

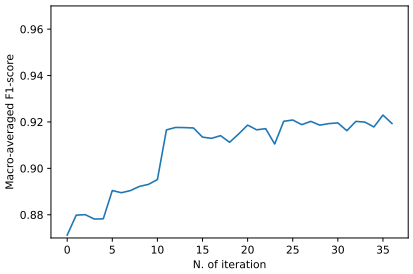

Figure 5: Macro-averaged F1-score trend.

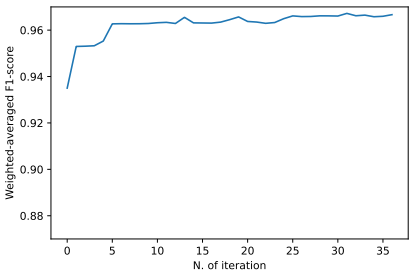

Figure 6: Weighted-averaged F1-score trend.

Given the outputs of our experiments (see Table 2), we can confirm that CRF-based tagging systems outperform HMM-based systems due to their application of feature functions, and that the feature selection optimization has a major impact on the system's overall performance.

Method	Macro F1	Weighted Avg
HMM	83.78%	91.32%
CRF only words	87.12%	93.50%
CRF best features	**92.30%**	**96.60%**

Table 2: Comparison among HMM's and CRFs' results.

6 Conclusion

In this paper, we develop a CRF-based POS tagger for English, focusing on a forward selection method that identifies the most impactful feature sets.

We have shown that optimizing the feature selection in CRFs for POS tagging tasks is possible and it can be of paramount importance in order to obtain a higher overall F1-score.

As future work, we would further test more features and experiment with multiple CRF methods and settings.

References

ACLWeb. 2019. POS Tagging (State of the art). Accessed: 2022-07-03.

Shriya Atmakuri, Bhavya Shahi, Ashwath Rao B, and Muralikrishna N. 2018. A comparison of features for pos tagging in kannada. *International Journal of Engineering Technology*, 7:2418–2421.

R. Harald Baayen and Rochelle Lieber. 1996. Word frequency distributions and lexical semantics. *Computers and the Humanities*, 30(4):281–291.

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

Daniel Jurafsky and James H Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR.

Maria Khalusova. 2019. Machine learning model evaluation metrics part 2: multiclass classification. Accessed: 2022-06-26.

Deepika Kumawat and Vinesh Jain. 2015. Pos tagging approaches: A comparison. *International Journal of Computer Applications*, 118(6).

Ruslan Mitkov. 2005. *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Mikael Nilsson. 2005. First Order Hidden Markov Model Theory and Implementation Issues. Technical report, Blekinge Institute of Technology, Department of Signal Processing.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Radoslaw Pytlak. 2009. *Conjugate Gradient Algorithms in Nonconvex Optimization*. Springer.

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

Sunil Ray. 2020. Introduction to feature selection methods with an example (or how to select the right variables?). Accessed: 2022-07-02.

Liam Stewart, Xuming He, and Richard S. Zemel. 2008. Learning flexible features for conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1415–1426.

Yang Sung-Hyun, Keshav Thapa, M Humayun Kabir, and Lee Hee-Chan. 2018. Log-viterbi algorithm applied on second-order hidden markov model for human activity recognition. *International Journal of Distributed Sensor Networks*, 14(4):1550147718772541.

Contributions

- Maria Vittoria Ateri: Baseline, Presentation Script, Most of Features and Feature Selection, Most of the Report, Report Trends, Final Testing.
- Malak Rassem: Baseline, Presentation Video and Editing, Some Feature Code, Forward Selection and CSV methods, User Input, Report Graphs, Report Editing.