UniBA @ KIPoS: A Hybrid Approach for Part-of-Speech Tagging

Giovanni Luca Izzi
University of Bari Aldo Moro
Department of Computer Science
via E. Orabona 4, 70125 Bari, Italy
giovannilucaizzi@gmail.com

Stefano Ferilli
University of Bari Aldo Moro
Department of Computer Science
via E. Orabona 4, 70125 Bari, Italy
stefano.ferilli@uniba.it

Abstract

English. The Part of Speech tagging operation is becoming increasingly important as it represents the starting point for other high-level operations such as Speech Recognition, Machine Translation, Parsing and Information Retrieval. Although the accuracy of state-of-the-art POS-taggers reach a high level of accuracy (around 96-97%) it cannot yet be considered a solved problem because there are many variables to take into account. For example, most of these systems use lexical knowledge to assign a tag to unknown words. The task solution proposed in this work is based on a hybrid tagger, which doesn't use any prior lexical knowledge, consisting of two different types of POS-taggers used sequentially: HMM tagger and RDRPOSTagger [(Nguyen et al., 2014), (Nguyen et al., 2016)]. We trained the hybrid model using the Development set and the combination of Development and Silver sets. The results have shown an accuracy of 0,8114 and 0,8100 respectively for the main task.

Italiano. *L'operazione di Part of Speech tagging sta diventando sempre più importante in quanto rappresenta il punto di partenza per altre operazioni di alto livello come Speech Recognition, Machine Translation, Parsing e Information Retrieval. Sebbene l'accuratezza dei POS tagger allo stato dell'arte raggiunga un alto livello di accuratezza (intorno al 96-97%), esso non può ancora essere considerato un problema risolto perché ci sono molte variabili da tenere in considerazione. Ad esempio, la maggior parte di questi sistemi utilizza della conoscenza linguistica per assegnare un tag alle parole sconosciute. La soluzione proposta in questo lavoro si basa su un tagger ibrido, che non utilizza alcuna conoscenza linguistica pregressa, costituito da due diversi tipi di POS-tagger usati in sequenza: HMM tagger e RDRPOSTagger [(Nguyen et al., 2014), (Nguyen et al., 2016)]. Abbiamo addestrato il modello ibrido utilizzando il Development Set e la combinazione di Silver e Development Sets. I risultati hanno mostrato un'accuratezza pari a 0,8114 e 0,8100 rispettivamente per il task main.*

1 Introduction

Part-of-Speech tagging (which we will shorten from now on with POS-tagging), as its name implies, is the operation of tagging each word with the corresponding part of the speech (POS-tag, from now on simply tag). Usually these tags are also applied to punctuation marks, such as commas, question marks and so on. POS-tagging models are essentials to build models for higher level operations. For example, they have been used to build Parsing Trees, which are used by Named Entity Recognition and Named Entity Linking systems to extrapolate entities starting from a document or short sentences. In this regard, we can't ignore that every day through social media a large amount of textual data are produced, these data present different structures and even different variants of the same language. Therefore, in this scenario the main requirement becomes the availability of highly reliable POS-tagging models capable of adapting to the different forms that a language can exhibit. Most

POS-tagging algorithms can be grouped into two classes: rule-based taggers and stochastic taggers. Rule-based taggers generally involve a large database of handwritten disambiguation rules that specify, for example, that a word with the ambiguous tag is a noun rather than a verb if it is preceded by a word that has "determiner" tag. While stochastic taggers generally solve the tagging ambiguities using a training set to calculate the probability that a given word has a given tag in a given context. There are also works that can be placed between these category like the Brill's works [(1992), (1994), (1995)]. However, most of these works include some lexical knowledge in order to tag word not learned during the training phase. It is a drawback we mustn't ignore because performance of these taggers may decrease, dramatically, for those languages where few or no lexical knowledge is available. Another important concern to think about are the necessary computational resources. For example (Mueller et al., 2013) reported that SVMTool tagger (Giménez et al., 2004) and CRFSuite tagger (Okazaki, 2007) require 2454 minutes (about 41 hours) and 9274 minutes (about 155 hours) respectively to complete the training phase on a dataset of 38727 sentences in the Czech language. The solution proposed in this work is a hybrid tagger whose philosophy is based on two simple factors: no use of lexical knowledge and no use of algorithms that require too high computational resources. For this reason we have decided to structure the hybrid tagger as a concatenation of a Hidden Markov Model (HMM) tagger and RDRPOSTagger [(Nguyen et al., 2014), (Nguyen et al., 2016)]. The proposed hybrid tagger has been evaluated during KIPoS task (Bosco et al., 2020) (KIParla Part of Speech) organized within Evalita 2020 (Basile et al., 2020), the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian, which will be held in Bologna (Italy) (December 16th – December 17th 2020).

KIPoS task consists of tagging a set of spoken sentences collected during some conversations held in Turin and Bologna. These conversations belong to different activity types: A1 (office hours), A3 (random conversation), C1 (exams), D1 (lessons) and D2 (interviews). A1, C1 and D1 are considered FORMAL conversations while A3 and D2 are considered INFORMAL conversations. Three different dataset were released: Development Set (DS), Silver Set (SS) and Test Set (TS). Each dataset is divided into Formal and Informal sentences, Table 1 show the details. The task is organized into three sub-tasks, based on the dataset used for training and testing the participants' systems:

- Main task - general: training on all given data (both DS-formal and DS-informal) and testing on all test set data (both TS-formal and TS-informal)
- Subtask A - crossFormal: training on data from DS-formal only and testing separately on data from formal register (TS-formal) and from informal register (TS-informal)
- Subtask B - crossInformal: training on data from DS-informal only and testing separately on data from formal register (TS-formal) and from informal register (TS-informal)

Dataset	**Conversation Turn**
DS-formal	1968
DS-informal	3383
DS+SS-formal	40768
DS+SS-informal	40817
TS-formal	455
TS-informal	571

Table 1: KIPoS datasets information

2 Description of the system

The proposed hybrid POS-tagger is a sequence of two POS-taggers, which don't use any prior lexical knowledge. We want to point out this sequence isn't fixed, anyone could create his one POS-tagger and replace one of the POS-tagger already used by the sequence. There is only one constraint that must be satisfied if you want to create a new tagger which will be the second tagger of the sequence, that is: the POS-tagger must be able to perform the learning starting from data tagged by the first tagger and perform the tagging operation on already tagged sentences. The first POS-tagger after receiving an untagged sentence (raw sentence) as input uses the information acquired during the training phase in order to transform this

sentence into a tagged sentence, where each token is associated with a tag according to the following structure token/tag. This first version of the tagged sentence could contain errors that will be corrected by the subsequent POS-tagger. The sequence implemented consist of an HMM tagger and a rule-based tagger called RDRPOSTagger. We believe these two POS-taggers can complement each other. Furthermore, RDRPOSTagger, unlike the other rules systems, is very light and allows to carry out the learning phase even if there are limited computational resources. Since the proposed solution doesn't use lexical knowledge, it allows us to have a model applicable to any language with homogeneous performance. Below we proceed with a brief description of the two POS-taggers.

2.1 HMM Tagger

In relation to POS-tagging there are many things to keep in mind when building an HMM tagger:

1. How to handle words not seen during the training phase?
2. How many previous tags should we consider?
3. How to handle the probability $P(t_i|t_{i-1})$ of a tag sequence not observed during the training phase?

A suffix-based approach is used in the HMM tagger designed to manage unknown words. Indeed, the suffixes are highly specific for each language and also they help to deduce the category to which the unknown word belongs. For example, in English the words ending in "-ing" may be gerunds or nouns. So the best strategy is to extract suffixes for each POS tag learned during the training phase. It is a fairly natural solution because for an HMM tagger it is necessary to keep, for each word, all the tags to which it can be associated and the number of times it has been associated with each single tag. To this purpose we keep, for each tag, a list of words where each word has been observed, during the training, associated to this tag. Finally, we extract a list of suffixes for each tag using the list of words mentioned before and a suffixes extraction algorithm. We developed the suffixes extraction algorithm using the Apriori Algorithm. The algorithm works as follow: Given a set of words *W*, in order to extract the candidate suffixes, first each word *w* is inverted, that is the letters that form the word are conversely listed starting from the last one up to the first letter. After doing this the set of inverted words is used as input to obtain a suffixes list containing lists of candidate suffixes of increasing size. Finally, the obtained suffixes list will be further processed to obtain a tree representation.The obtained tree will be cut considering, at every node, three different thresholds: the support of this node, the number of distinct words which contain the suffix represented by this node, the percentage of *W* words which contain the current suffix and the suffix from which it is derived.
Regarding the second question, in the planned HMM tagger it was decided to consider the trigrams, that is for each tag the two previous tags are considered. Then the transition probability becomes: $P(t_i|t_{i-1}, t_{i-2})$. Considering trigram-based transition probabilities is the most commonly used method in state-of-the-art stochastic POS-taggers. At this point also the last question changes, since we are now interested in solving problems deriving from sequences of trigrams not observed during the training phase. The approach used to manage unknown tag sequences is a smoothing technique called linear interpolation described by the following formula:

$$P(t_i|t_{i-1}, t_{i-2}) = \lambda_3 P_{MLE}(t_i|t_{i-1}, t_{i-2}) + \lambda_2 P_{MLE}(t_i|t_{i-1}) + \lambda_1 P_{MLE}(t_i)$$

The main requirement of this formula is $\lambda_1 + \lambda_2 + \lambda_3 = 1$, thus ensuring that *P* is a probability distribution. The λ values are learned using the deleted interpolation (Jelinek et al., 1980), where we subsequently delete each trigram from the training dataset and choose the λ in order to maximize the probability of the rest of the dataset.

2.2 RDRPOSTagger

RDRPOSTagger [(Nguyen et al., 2014), (Nguyen et al., 2016)] is a rule-based tagger, this approach is also called transformation-based error-driven, able to automatically structure the rules in a particular tree structure called Single Classification Ripple Down Rules (SCRDR) [(Compton and Jansen, 1990), (Richards, 2009), (Nguyen et al., 2015)]. A SCRDR tree is a binary tree with two distinct

Training dataset	Formal (F)	num KF	KF	num UF	UF
DS	0.8236	2940	0.9180	638	0.3887
DS-formal	0.7954	2769	0.9176	809	0.3770
DS-informal	0.7778	2805	0.8709	773	0.4398
DS+SS	0.8085	3429	0.8293	149	0.3288
DS+SS-formal	0.8113	3406	0.8352	172	0.3372
DS+SS-informal	0.7758	3190	0.8128	388	0.4716

Table 2: Results obtained for Gold Test corrected Formal sentences

types of edges. These edges are usually called: except and if-not. Each tree node corresponds to a rule. Each rule has the form: if $\alpha \rightarrow \beta$, where α is the condition of the rule and β is the conclusion. Cases in a SCRDR tree are evaluated by passing a case to the root of the tree. In each node of the tree, if the condition of the rule in a node η is satisfied by the input case (so the node η is activated), the case is passed to the node except child of the node η using the except edge if it exists. Otherwise, the case is passed to the if-not node child of the node η. The conclusion of this process is given by the last activated node. A new node containing a new exception rule is added to a SCRDR tree when the evaluation process returns a wrong conclusion. The new node is connected to the last node in the evaluation path of a given case through an except edge if the last node of the path is the activated node, otherwise, it is connected to it with an if-not edge. To ensure that a conclusion is always provided, the root node (called the default node) generally contains a trivial condition that is always satisfied. The rule in the default node, called the default rule, is the only rule that is not the exception rule of any other rule. We decided to use RDRPOSTagger as a second tagger of our sequence because of its own abilities: It is a lightweight rule tagger; rules are learned in a controlled context, in this way they can't influence one another. Therefore, our hybrid model is very fast during training and tagging phase.

3 Results

We evaluated the performance of the hybrid tagger just described with just a single run as we did for the competition. For the competition we used only the DS dataset for the learning phase, but here we investigated experimental results using also the SS dataset. More precisely, we used Random Split to divide the dataset into 90% training set and 10% validation set, the latter has been used to learn the rules through RDRPOSTagger. We decided to use default configuration for RDRPOSTagger and for our suffixes extraction algorithm. More precisely we set the three different thresholds described before equals to 10, 3 and 0.4 respectively. Table 2 and Table 3 show the results obtained for Formal and Informal corrected Gold Test dataset, provided by the authors after the evaluation, which contains some improvements compared to the test dataset used during the competition. In these two tables we present the results listing: overall accuracy, number of known tokens, known tokens accuracy, number of unknown tokens and unknown tokens accuracy.

4 Discussion

The test dataset provided for the competition contains spoken sentences based on conversation turns, which make the competition quite challenging because these sentences have an irregular structure with misspelled words. Our evaluation will also have to take into account the number of conversation turns contained in the training dataset, fewer conversation turns in the overall dataset will imply fewer conversation turns in the validation set and therefore fewer rules learned by RDRPOSTagger. In fact, using only the DS it is able to learn about 5-6 rules while on the combination of DS-SS the rules learned are about 40. Moreover, these rules depend on the contexts contained in the validation set which, given the small number of data, can be very different from those encountered during the testing phase. Abstracting from the number of known words, which increase using the combination of the two datasets, the results show that the accuracy on these words remains around 90% when learning is performed using the Development Set (DS). While using the combination of Silver (SS) and Development Sets this percentage is closer to 80% and it is surprising if we consider that the DS contains far less data.

Training dataset	Informal (I)	num KI	KI	num UI	UI
DS	0.7992	3213	0.8954	587	0.2725
DS-formal	0.7631	3026	0.8853	774	0.2855
DS-informal	0.7802	3128	0.8772	672	0.3288
DS+SS	0.8425	3602	0.8425	198	0.2474
DS+SS-formal	0.7821	3436	0.8378	364	0.2554
DS+SS-informal	0.8050	3555	0.8447	245	0.2285

Table 3: Results obtained for Gold Test corrected Informal sentences

Such a difference can be explained if we consider that SS is an automatically tagged dataset and it isn't manually revised so it can be source of errors. Only for the subB task the accuracy on unknown words, considering a formal context, exceed, even if slightly, the results obtained using the DS. The results for the unknown words are quite low, these errors in turn propagate other errors on the known words. The errors concern words that are impossible to recognize without the use of lexical knowledge such as names, they are also written with a lowercase initial, date and numbers written in textual format. Other errors are related to polysemy words such as the word "prego" used as both INTJ and VERB. However, in this case the word has been observed during training more often as VERB than INTJ and the particular contexts of the test sentences and those learned during training don't help us to tend towards the correct INTJ tag.

5 Conclusion

The KIPoS competition was the perfect situation to evaluate the solution we proposed because there are formal and informal sentences and they don't have a regular structure. In this work we presented a hybrid POS-tagger that tries to combine the advantages of a stochastic model and a rule model without using previous lexical knowledge while keeping learning and tagging times at a level suitable for real applications. Results showed that the percentage of known words tagged correctly is about 90% while for the unknown words the percentages vary in the range [27% - 44%], where the extremes of this interval represent the worst and best case respectively. The greatest difficulties occurred for unknown words in the informal context. The competition allowed us to get useful insights regarding which parts of the system need to be improved. For example, our suffixes extraction algorithm, which is still in a beta version. Future work directions will surely focus on improving the suffixes extraction algorithm and on the possible combination of suffixes and prefixes to identify the unknown words. Every future directions will always investigate solutions which will not require lexical knowledge. Therefore, they will be applicable to any language.

References

Bosco, Cristina and Ballarè, Silvia and Cerruti, Massimo and Goria, Eugenio and Mauri, Caterina. 2020. *KIPoS@EVALITA2020: Overview of the Task on KIParla Part of Speech tagging*. Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020).

Basile, Valerio and Croce, Danilo and Di Maro, Maria, and Passaro, Lucia C. 2020. *EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*. Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020).

Eric Brill. 1992. *A simple rule-based part of speech tagger. In Proceedings of the third conference on Applied natural language processing (ANLC '92)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 152-155. DOI: `https://doi.org/10.3115/974499.974526`

Eric Brill. 1994. *Some advances in transformation-based Part of Speech tagging*. Proceedings of the 12th National Conference on Artificial Intelligence (AAAI) vol. 1, pages 722–727.

Eric Brill. 1995. *Unsupervised learning of disambiguation rules for Part of Speech tagging*. In Natural Language Processing Using Very Large Corpora Workshop, pages 1–13. Kluwer.

T. Mueller, H. Schmid, and H. Schütze. 2013. *Efficient Higher-Order CRFs for Morphological Tagging*. In Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing, pages 322– 332.

J. Giménez, L. Màrquez, and L. Marquez. 2004. *SVMTool: A General POS Tagger Generator Based on Support Vector Machines*. In Proceedings of the 4th International Conference on Language Resources and Evaluation, pages 43–46.

N. Okazaki. 2007. *CRFsuite: A Fast Implementation of Conditional Random Fields (CRFs)*. http://www.chokkan.org/software/crfsuite/

Nguyen, Dat Quoc and Nguyen, Dai and Pham, Dang and Pham, Son. 2014. *RDRPOSTagger: A Ripple Down Rules-based Part-Of-Speech Tagger*. 17-20. 10.3115/v1/E14-2005.

Nguyen, Dat Quoc and Nguyen, Dai and Pham, Dang and Pham, Son. 2016. *A Robust Transformation-Based Learning Approach Using Ripple Down Rules for Part-of-Speech Tagging*. AI Communications. 29. 409-422. 10.3233/AIC-150698.

Jelinek, F. and Mercer, R. L. 1980. *Interpolated estimation of Markov source parameters from sparse data*. In Gelsema, E. S. and Kanal, L. N. (Eds.), Proceedings, Workshop on Pattern Recognition in Practice, pp. 381–397. North Holland.

P. Compton and R. Jansen. 1990. *A Philosophical Basis for Knowledge Acquisition*. Knowledge Acquisition, 2(3): 241–257.

D. Richards. 2009. *Two Decades of Ripple Down Rules Research*. Knowledge Engineering Review, 24(2):159–184.

D. Q. Nguyen, D. Q. Nguyen, and S. B. Pham. 2015. *Ripple Down Rules for Question Answering*. Semantic Web journal, to appear, 2015. URL: http://www.semantic-web-journal.net/