Ensemble system for Part-of-Speech tagging

Felice Dell'Orletta

ILC-CNR - via G. Moruzzi 1
56100 Pisa, Italy
felice.dellorletta@ilc.cnr.it

Abstract. The paper contains a description of the Felice-POS-Tagger and of its performance in Evalita 2009. Felice-POS-Tagger is an ensemble system that combines six different POS taggers. When evaluated on the official test set, the ensemble system outperforms each of the single tagger components and achieves the highest accuracy score in Evalita 2009 POS Closed Task. It is shown first that the errors made from the different taggers are complementary, and then how to use this complementary behavior to the POS tagger's advantage.

Key words: POS tagging, Hidden Markov Model, Support Vector Machine, Maximum Entropy.

1 Introduction

Part-of-speech tagging is a very important step in natural language processing (NLP) and in most advanced language technology systems. Although the high accuracy scores of state of the art POS taggers (for Italian is between 97% and 98% [9], for only lexical categories and using a morphological lexicon), POS tagging remains a central problem because is typically the first step of NLP Pipeline Architectures. Therefore a POS tagging error may cause error propagation to the following steps of natural language processing. For instance Watson [11] and Yoshida et al. [5] show the impact of part-of speech errors in parsing task.

In EVALITA 2009 Part-of-Speech Closed task the POS tagging involves the assignment of both lexical category and morphological features to each token. The task is a *closed* task which means that the taggers cannot use any external resources besides the supplied official training, development and test sets (Tanl tagset [8]).

This paper contains the description of our participation to the task.

2 Component taggers

The Felice-POS-Tagger is a combination of six component taggers, with three different algorithms, each of which is used to construct a left-to-right (LR) tagger and a right-to-left (RL) tagger.

The first POS tagging algorithm is a popular algorithm for tagging based on Trigrams'n'Tags (TnT) [10] which has readily available open source reimplementation called *Hunpos* [7]. TnT is based on the Viterbi algorithm for second order Markov models. TnT uses various methods of smoothing and of handling unknown words. In particular, the main paradigm used for smoothing is linear interpolation and Unknown words are handled by a suffix trie and successive abstraction. [10] shows a detailed description of the techniques used in TnT.

The other two tagging algorithms are based on ILC-UniPi-tagger [3]. We developed a modular python implementation of this tagger that can use several learning algorithms and provides a simple definition of feature models, through a configuration file (work influenced by the study done by Attardi for the development of the DeSR [4] dependency parser). In the context of the Evalita 2009 POS tagging task we used Support Vector Machines (SVM) (LIBSVM [2] package) and Maximum Entropy (ME) (MAXENT [6] package) as learning algorithms. Tables 1, 2 and 3 show the feature models for the SVM and the ME taggers. Right-to-left and left-to-right taggers use the same set of features. Token 0 is the current token being analyzed, positive and negative numbers are respectively the successive and the previous tokens in the input sentence. *FORM* is the word form or punctuation symbol, *FORM_LENGTH* is the length (in characters) of the word form, *FORM_FORMAT* and *FORM_SHAPE* capture the orthographic properties of the analyzed word form, *FORM_PREFIX* and *FORM_SUFFIX* are all the prefixes and suffixes of the word up to a configurable maximum length (five characters for these experiments), *POS* is the part of speech. Only for ME-based taggers we use bigram and trigram features (table 3).

Table 4 reports the relative accuracies of the six component taggers.

Most of the time was spent designing and developing the software, which limited the time alloted for optimizing learning algorithm parameters and for selecting the best set of features for each parser. This means that the performances of the two SVM and the two ME taggers can probably be improved.

Table 1. SVM (RL and LR) Felice-POS-Tagger: feature models.

Feature	token
FORM	-2 -1 0 1
FORM_LENGTH	0
FORM_FORMAT	0
FORM_PREFIX	0
FORM_SUFFIX	0
FORM_SHAPE	0
POS	-1

Table 2. ME (RL and LR) Felice-POS-Tagger: feature models.

Feature	token
FORM	-2 -1 0 1
FORM_LENGTH	0
FORM_FORMAT	0
FORM_PREFIX	0
FORM_SUFFIX	0
FORM_SHAPE	0
POS	-1

Table 3. ME (RL and LR) Felice-POS-Tagger: bigram and trigram features. (W_x=form of token x ; P_x=POS of token x)

BIGRAM	$(P_{-1}W_0)$ $(W_{-1}W_0)$ (W_0W_1) (W_1W_2)
TRIGRAM	$(P_{-2}P_{-1}W_0)$ $(W_{-1}W_0W_1)$ $(W_{-2}W_{-1}W_0)$ $(W_0W_1W_2)$

Table 4. Accuracy of the component POS taggers

	H-LR	H-RL	SVM-RL	SVM-LR	ME-RL	ME-LR
development set	92.82	92.72	91.39	91.16	91.19	90.84
test set	95.97	95.55	94.76	94.29	94.25	93.76

3 Complementarity, Disagreement and Additivity rates

In this section we want to show that the errors the different taggers make are complementary. It's clear that if all the taggers made the same errors or if the lower accuracy tagger errors contain all the higher accuracy tagger errors, the tagger would have not improved accuracy through classifier combination. To see if the component taggers used are complementary, we show a series of evaluation measures proposed by Brill and Wu in [1] to calculate how different the errors of the taggers are (evaluation on Evalita-2009 test set).

Brill and Wu define the *complementary* rate of taggers A and B as:

$$Comp(A, B) = (1 - \frac{\#of\ common\ errors}{\#of\ errors\ in\ A\ only}) * 100$$

$Comp(A, B)$ measures the percentage of time when tagger A is wrong and that tagger B is correct. Table 5 shows the complementary rate between the different taggers. As is shown, although the two hunpos taggers are fairly more accurate than the others, their errors are quite complementary with respect to the other four taggers. For instance, when the Hunpos left-to-right (H-LR) tagger is wrong, the worst tagger (ME-LR) is correct 36.36% of the time. In addition, Table 5 shows that left-to-right and right-to-left taggers are quite complementary. For instance, when the SVM-RL tagger is wrong the SVM-LR tagger is correct 34.16% of the time.

Table 5. Complementarity rate. Comp(A,B). Row=A, Column=B

	H-LR	H-RL	SVM-RL	SVM-LR	ME-RL	ME-LR
H-LR	0	15.66	33.84	36.36	34.85	36.36
H-RL	23.74	0	34.70	35.16	37.90	39.27
SVM-RL	49.22	44.57	0	28.29	26.74	32.95
SVM-LR	55.16	49.47	34.16	0	40.93	29.18
ME-RL	54.42	51.94	33.22	41.34	0	31.80
ME-LR	58.96	56.68	43.65	35.18	37.13	0

Table 6 shows the *disagreement* rate between the different taggers. The Disagree score for a component tagger A measures the percentage of time that tagger A disagrees with at least one of the other taggers and A is wrong.

Quoting Brill and Wu [1]:

> A tagger is much more likely to have misclassified the tag for a word in instances where there is disagreement with at least one of the other classifiers than in the case where all classifiers agree.

It is interesting to note that when the best tagger (H-LR) disagrees with the others the Hunpos-LR error rate is 29.70%, instead of the overall error rate 4.03%.

Table 6. Disagreement rate

	H-LR	H-RL	SVM-RL	SVM-LR	ME-RL	ME-LR
Overall Error Rate	4.03	4.45	5.24	5.71	5.75	6.24
Error Rate When Disagreement	29.70	34.02	42.06	46.80	47.21	52.16

The table 7 shows that the tagger complementarity is *additive*. The first row in the table is the additive error rate of an oracle that can choose among all of the possible outputs of component taggers. The second row is the additive oracle improvement. As it is shown, if the oracle uses all the six taggers the additive error rate is 1.74 %, (which means) a decrease of 56.57% with respect to the best tagger (4.03%).

After these analyses, we can conclude that it may be possible to obtain an improvement of the accuracy in POS tagging when combining the six component taggers.

Table 7. Additivity rate

	H-LR	+H-RL	+SVM-RL	+SVM-LR	+ME-RL	+ME-LR
% of time all are wrong	4.03	3.39	2.38	2.11	1.85	1.74
% Oracle Improvement		15.66	40.91	47.47	54.04	56.57

4 Taggers Combination

Felice-POS-Tagger can combine the outputs of the component taggers using three different methods:

- Simple Voting scheme;
- machine-learning classifier to identify the correct output among the outputs of the component taggers;
- machine-learning classifier to identify the correct POS tag using the outputs of component taggers as features.

Experiments conducted on Evalita-2009 development set showed that using the two machine-learning classifier methods we do not achieve an improvement in accuracy score compared to the best single tagger, or very slight improvements are obtained. Both SVM and ME machine-learning algorithms have been used for the combination experiments and the training set was created using a tenfold method: the original training set was splitted into ten parts and for each part we have trained the component taggers on the other parts and then we tagged the excluded one. At the end of this process we obtained the training set for combination methods. It is important to emphasize that the time for testing the machine-learning classifier methods was really short therefore we probably don't use the best feature models in our experiments.

We achieve the best accuracy score using the simple voting scheme method. This method consists in combining the outputs of the six individual taggers, choosing for each token the part-of-speech that is selected from the largest numbers of taggers. In case of ties between two or more part-of-speeches we choose the one predicted from the best individual model. Table 8 shows the accuracy scores of the simple voting combination Felice-POS-Tagger for development and test sets.

Table 8. Accuracy scores for development and official test sets

	development set	test set
H-LR	92.82	95.97
Voting Combination	93.24	96,34
% error rate reduction	6.27	9.09

As we can see, the Simple Voting allows us to obtain an improvement of 0.42% on the development set and 0,37% on the test set. That is, respectively, a relative error rate reduction of 6.27% and 9.09%, relative to the accuracy of the best single tagger (H-LR).

5 Conclusion

In this paper we report our participation to the EVALITA 2009 Part-of-Speech Closed task. Our tagger, Felice-POS-Tagger, achieves the best score of the competition.

In this work, most of the time was spent designing and developing the software, which limited the time alloted for optimizing learning algorithm parameters and for selecting the best set of feature models. For this reason, future works should be dedicated to the selection of new feature models in order to improve the accuracy scores of single component taggers and final ensemble systems. Moreover further methods of combination should be studied.

References

1. Brill, E., WuSmith, J.: Classifier Combination for Improved Lexical Disambiguation. In: Proceeding of COLING-ACL 1998, pp. 191–195 (1998)
2. Chih-Chung, C., Chih-Jen, L.: LIBSVM: a library for support vector machines, `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2001)
3. Dell'Orletta, F., Federico, M., Montemagni, S., Pirrelli, V.: Maximum Entropy for Italian POS Tagging. In: Proceedings of Workshop Evalita 2007. Inteligenza Artificiale, vol. 4, issue 2 (2007)
4. DeSR: a Multilingual Shift-Reduce Dependency Parser, `http://www.desr.org/`
5. Yoshida, K., Tsuruoka, Y., Miyao, Y., Tsujii, J.: Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In: Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07). Hyderabad, India (2007)
6. MAXENT, `http://sourceforge.net/projects/maxent/`
7. Hunpos, `http://code.google.com/p/hunpos/`
8. Tanl: Text Analytics and Natural Language processing, Project Analisi di Testi per il Semantic Web e il Question Answering, `http://medialab.di.unipi.it/wiki/SemaWiki` (2008)
9. Tamburini, F.: Evalita 2007: The Part-of-Speech Tagging Task. In: Proceedings of Workshop Evalita 2007. Inteligenza Artificiale, vol. IV, issue 2 (2007)
10. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000. Seattle, WA (2000)
11. Watson, R.: Part-of-speech tagging models for parsing. In: Proceedings of the 9th Annual CLUK Colloquium. Open University, Milton Keynes, UK (2006)