Bidirectional Sequence Classification for Part of Speech Tagging

Andrea Gesmundo

Univ Geneva, Dept of Computer Science and Dept of Linguistic,
route de Drize 7, 1227 Carouge, Switzerland
andrea.gesmundo@unige.ch

Abstract. With this paper is presented a system for Part of Speech Tagging, based on the Perceptron Algorithm. In the proposed framework, the order of the inference is not forced into a monotonic behavior (left-to-right), but is learned together with the parameters of the local classifier. The system tested on the task of Italian POS Tagging at EVALITA 2009 obtained the second position, with a Tagging Accuracy of 95.82%.

Key words: Guided Learning, Perceptron, POS Tagging.

1 Introduction

Part of Speech Tagging (POS) is a subtask of Natural Language Processing. The goal of this task is to assign a label to each word in the text, this label consists of a combination of lexical and morphological features. The system described in this paper carries out POS tagging experiments with semi-supervised training. In particular we extend to the Guided Learning (GL) framework presented in [1]. This approach is more complex than supervised learning. The system can learn the parameters for the local classifier from gold standard labels, but has no indications on the order of inference.

Compared to others approaches, GL shows some advantages, it does not suffer from the label bias problem [2]. Basing the learning algorithm on the Perceptron scheme allows one to keep a low system complexity and moderate execution time, without sacrificing learning capability and quality of the results. With regard to others systems that use a Perceptron algorithm, like [3], GL introduces a bidirectional search strategy. Instead of forcing the order of the tagging in a left-to-right fashion, any tagging order is allowed. It follows a easiest-first approach and incorporates the learning of the order of inference in the training phase. In this way right-context and bidirectional-context features can be used at little extra cost.

As shown by the results obtained in Evalita 2009 POS Shared Tasks and in [1] [5] and [6], GL is a framework that can be adapted to a variety of tagging tasks, ensuring state of the art results and short training times.

2 Bidirectional Guided Classification

In this section we present the Inference Algorithm and the Training Algorithm.

2.1 Inference Algorithm

As input to the Inference Algorithm we have a sequence of tokens $t_1 t_2 \cdots t_n$. For each token t_i, we have to assign a label $l_i \in L$, with L being the label set. A subsequence $t_i \cdots t_j$ is called a span, and is denoted $[i, j]$. To each span s are associated one or more hypotheses, composed by a sequence of length $|s|$ over L.

The labels located at the boundaries of an hypothesis sequence are used as context for labeling tokens outside the span s. In our case a trigram model is used, so when choosing the label for the token t_i we can use the two boundary labels (l_{i+1}, l_{i+2}) of the right span $[i+1, j]$ if this has already been tagged. Similarly, we can use the two labels (l_{i-2}, l_{i-1}) as left context for the current tagging operation in the case that the left span $[k, i-1]$ is available. We will refer to the left two label as the left interface I_{left}, and to the right two labels as the right interface I_{right}.

We denote the boundaries of a span s with $b = (I_{left}, I_{right})$, b contains the labels relevant for the tagging of neighboring tokens. We partition the hypotheses associated with span s into sets compatible with the same boundaries b. For each span s we use a table M_s indexed by all possible b, so that $M_s(b)$ is the set of all hypotheses associated with s that are compatible with I_{left} and I_{right}.

For a span s, we denote the associated top hypothesis with:

$$h_s^* = \underset{h \in M_s(b), \forall b : M_s(b) \neq \emptyset}{\operatorname{argmax}} V(h) \tag{1}$$

where V is the score function of a hypothesis.

Spans are started and grown by means of tagging actions. Three kinds of actions are available: it is possible to start a new span by labeling a token with no context, or expand an existing span by labeling an adjacent token, or merging two spans by labeling the token between them. In this last case the two originating spans would be subsequences of the resulting span, and the labeling action of the token between the spans will use both right and left context information.

For each hypothesis h associated with a span s, we maintain its most recent tagging action $a(h)$, and the hypotheses, if any, that have been used as left context $h_L^*(h)$ and right context $h_R^*(h)$.

We can now define the score function for hypotheses in a recursive fashion:

$$V(h) = V(h_L^*(h)) + V(h_R^*(h)) + U(a(h)) \tag{2}$$

The score of the current tagging action $U(a(h))$ is added to the score of the top hypotheses that might have served as context and have been merged in the new hypothesis. The score of the labeling action $U(a(h))$ is computed through a linear combination of the weight vector w and the feature vector of the action $f(a(h))$:

$$U(a(h)) = w \cdot f(a(h)) \tag{3}$$

To reduce the search space explored during the inference algorithm we apply a beam search strategy. The beam width B determines the maximum number of boundaries b maintained for each span s. The value of B is given as input, as the weight vector w used to compute the score of an action.

The algorithm works using two groups of spans: P is the list of accepted spans, and Q is the a queue of candidate spans. Q can contain new spans of length one or extension of spans previously accepted and at the current time located in P.

At the beginning of the inference algorithm P is initialized with the empty set, and Q is filled with candidate spans $[i, i]$ for each token t_i, and for each label $l \in L$ assigned to t_i we set:

$$M_{[i,i]}((l, l)) = \{i \to l\} \tag{4}$$

where $i \to l$ represent the hypothesis consisting of the action with no context which assigns label l to w_i. This provides the set of starting hypotheses.

The loop of the algorithm repeatedly selects a candidate span s' from Q, s' is the span with the highest action score, so we pick the span that represents the next tagging action we are most confident about.

Now we use s' to update P and Q. First we update P, adding s' and removing the spans included in s'. Then let S be the set of spans removed from P. We update Q removing each span which takes one of the spans in P as context, and replace it with a new candidate span taking s' as new context.

The algorithm terminates when P contains a single span covering the whole token sequence and Q becomes empty.

The loop is guaranteed to terminate since at each iteration a span is expanded or added in P, and considering that P cannot have overlapping spans we can conclude that the number of iterations needed is linear with the size of the token sequence.

2.2 Learning Algorithm

In this section we describe the Guided Learning Algorithm, used to learn the weight vector w with a Perceptron-like algorithm.

For the training a set of token sequences $\{T_1, T_2, \cdots, T_m\}$ is provided as input. to each token sequence $T_r = (t_1, t_2, \cdots t_n)$ is paired a gold standard label sequence of the same length $L_r = (l_1, l_2, \cdots, l_n)$. At the beginning of the learning algorithm we initialize P and Q as we do in the inference algorithm. Then we iterate selecting the candidate span s' for the next labeling action from Q like in the inference algorithm. If the s' top hypothesis match on the gold standard, we update P and Q as in the inference algorithm. Otherwise, we update the weight vector w by promoting the features of the gold standard, and demoting the features of the action of the candidate top hypothesis, like in the Perceptron algorithm. Then we undo the last labeling action by replacing the elements in Q with a new list of candidates containing all the possible spans based on the context spans in P, and computing the new scores with the updated weight vector w.

In our implementation we have used the Averaged Perceptron [3] and Perceptron with margin [4].

3 Experiments

In this section we are going to describe the setting chosen for the final experiment of the Evalita 2009 POS tagging task, we also report and discuss the results.

3.1 Setting

In the setting of our best system we set beam width $B = 3$, as threshold between speed and accuracy.

Among the set of features used, we distinguish between context features and lexical features. Context features are meant to capture the information of the surrounding words and labels, while lexical features concern the form of the current word and possibly its relation with lexical characteristics of the context words.

Context Features : To exploit the bidirectional context window over the labels and words we adopted a feature set that already has given state of the art results in POS tagging task on others corpora. We report the feature templates in Table 1 .

Table 1. Templates for context features: 1) single word features, 2) couple of words features, 3) left context features, 4) right context features, 5) and bidirectional features.

1	$[w_0]$, $[w_{-2}]$, $[w_{-1}]$, $[w_1]$, $[w_2]$
2	$[w_{-1},w_0]$, $[w_0,w_1]$
3	$[p_{-1}]$, $[p_{-2},p_{-1}]$, $[p_{-2},p_{-1},w_0]$, $[p_{-1},w_0]$, $[p_{-2}]$, $[p_{-2},w_0]$
4	$[p_1]$, $[p_1,p_2]$, $[p_1,p_2,w_0]$, $[p_1,w_0]$, $[p_2]$, $[p_2,w_0]$
5	$[p_{-1},p_1]$, $[p_{-1},p_1,w_0]$

We made attempts to modify this set of features, removing features subsets and trying to find new feature patterns with semi-automatic techniques. Unfortunately these attempts lead to no improvements. This made us believe that this set of features is general enough for this kind of task, and preserves its effectiveness with corpora in different languages.

Considering the fine grained structure of the tag-set used, we introduced a new kind of context feature that considers just the prefix of the actual POS tag, excluding the morphological information encoded in the last part of the label. After trying different settings we obtained the best improvement using the label prefixes for a window of size three centered in the current word. This led to a 3% relative error reduction.

Lexical Features : As lexical features for the current word we consider: the presence of special characters of symbols like digits or '-'; the prefixes and suffixes up to length of 9 characters; the capitalization of the first letter or of the whole word, in relation with the capitalization of context words.

About the capitalization lexical feature, we observed that treating the first word of the sentence separately from the other words of the sentence lead to an improvement of the performances. This resulted in a relative error reduction of 0.9%.

3.2 Results

The results reported in this section are those obtained on the Evalita 2009 data set, this corpus is composed of articles from an Italian newspaper, tagged with the Tanl tag-set consisting of 328 tags, grouped in 14 basic categories.

The corpus was distributed divided in 3 segments, a "train" set (3,719 sentences) used for training the system, a "devel" set (147 sentences) to be used as for development, and a "test" set (147 sentences) provided with no labels and to be tagged for the final evaluation.

The participating systems are evaluated on the tagging accuracy (TA) computed on the submitted test set. The unknown words tagging accuracy (UWTA) it is also considered. The results for the first three systems in the final rank are reported in Table 2.

Table 2. First three systems classified in the Evalita 2009 POS tagging task, our Guided Learning based system ranked second.

	TA	UWTA
system A	96.34%	91.07%
Guided Learning	95.85%	91.41%
system C	95.73%	90.15%

We can observe that the first three systems reached a similar score, we can consider this value to be the state of the art for this corpus. The system C just made 30 more labeling errors than the system A, out of a total of 4919 tags.

We can also notice that the Guided Learning approach obtained the best result on the UWTA. This is probably a consequence of the lack of constraints in the inference sequence. As explained earlier, the Guided Learning approach follows a tagging order based on an easiest-first heuristic. This method postpones difficult decisions (as in the case of unknown words), and assigns a label when more context information is available.

Moreover we observe that the error rate obtained on this POS corpus is noticeably higher then the error rate obtained on others POS corpora. As example we cite [1], where an earlier version of this system reached the state of the art TA of 97.33% on the Wall Street Journal corpus tagged with Penn Treebank tag-set. We think that this gap is not due to the difference of language, but rather it can be caused by the use of a far larger tag-set (that brings more uncertainty during the prediction phase), or can be due to the presence of higher internal noise of the corpus.

Since the system relies on a simple, but effective inference algorithm and the training algorithm is based on the averaged Perceptron (known for the speed of its implementations), we were able to record short execution times despite the large tag-set. On a common Desktop (equipped with a Core2 Duo CPU at 2.66GHz) the 20 rounds of training were completed in 12 hours, and the prediction of the test set was done in 2 minutes. During the training phase 1M features were generated.

4 Conclusion

In this paper we extended the work on the Guided Learning approach, adapting it to a new task, and applying new features. We successfully participated at the Evalita 2009 POS shared task, achieving the second position in the final rank. The evaluation results show that a state of the art tagging accuracy is reached. Furthermore the system obtained the best score in the unknown words tagging accuracy showing the effectiveness of the GL approach of dynamically incorporating the order of inference and the local classification in the learning phase. In related works, described in [1] [5] and [6], we applied this approach to a variety of tasks (POS tagging, NER, NP chunking) reaching state of the art results with moderate execution time. With this work, we have further proved the validity of Guided Learning.

Acknowledgments. We thank Giorgio Satta and Libin Shen for help and advice.

References

1. Shen, L., Satta, G., Joshi, A., K.: Guided learning for bidirectional sequence classification. In: Proceedings of the 45th annual meeting of the association of computational linguistics (2007)
2. Bottou, L.: Une approche théorique de l'apprentissage connexionniste: Applications à la reconnaissance de la parole. Ph.D thesis, Université de Paris XI (1991)
3. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: EMNLP-2002 (2002)
4. Krauth, W., and Mézard, M.: Learning algorithms with optimal stability in neural networks. Journal of Physics A, 20:745-752 (1987)
5. Gesmundo, A.: Elaborazione del linguaggio naturale basata su features bidirezionali. Master thesis, Università di Padova (2007)
6. Gesmundo, A.: Bidirectional Sequence Classification for Named Entities Recognition. In: EVALITA-2009 (2009)