PoSTagger Based on SVM

Leonardo Rigutini and Guido Gioioso

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Siena

Abstract. In this paper we present a Pos Tagging system based on Support Vector Machines. The system is based on a sliding-window approach in which the window moves from left-to-right. It exploits a rich set of features including morphological features of the word under analysis and of a fixed set of surrounding words. The system joined the Italian Pos Tagging task of Evalita2009 scoring an accuracy of 93,37% on the test set provided by the organization.

Keywords: SVM, PoS Tagging.

1 Introduction

The Part of Speech tagging is the problem of determining the correct grammatical property of a word such as: noun, verb, adjective and so on. Furthermore if a tagger select a "noun" tag for a word, it must specify the number and the gender or the time in the case of a "verb" tag. The system described in this article use a Support Vector Machine [2, 1] to learn grammatical rules providing a state-of-the-art results with little tuning.

2 System Description

The PoSTagger scans the "taggable" text using a sliding-window with a customizable size. In this description, let be w_i the word under analysis and let be $\omega = 2k + 1$ the window size. The tag of the i^{th} word w_i is predicted using information about $w_{i-k} \cdots w_{i+k}$ words. A set of binary features is computed for each word in the window. The system allows to insert in the pattern the following features for each word:

Word features: these features represent some properties concerning the raw characters of the word. They account, for example, for the case of the first and the other characters of the word, or for presence of numeric or non-alpha-numeric symbols (like punctuation symbols) in the word.

PoS Features: all the words in the window preceding that one under analysis w_i are already been tagged; their tags can be used as a features of central

word of the window. Let be $Tpos = \{t_1, t_2, t_3, \cdots, t_p\}$ the list of possible tags assignable to a word. For each word in $w_{i-k} \cdots w_{i-1}$ a set of p binary features are inserted in the pattern. For each set, only the value of one bit is 1 and, in particular, the j^{th} bit of the set is 1 if t_j is the tag assigned by the system to that word.

Dictionary Features: each word in the training set is inserted into a dictionary V. The presence of a word in the dictionary is a feature that we represent in the pattern with a $|V|$-dimensional vector of bits where the j^{th} bit is 1 if the word-id in V is j and 0 otherwise.

Prefix and Suffix Features: the system allows to represent in the pattern associated to a word, some features concerning the prefix and the suffix of the word. Is possible to represent prefixes and suffixes of different size for the same word. Let be $A = \{a_1, a_2, a_3, \cdots\}$ the alphabet containing all possible characters of a word. To represent a k-characters suffix we need to introduce k sets of $|A|$ bits in the pattern. For each set, the value of the j^{th} bit is 1 if the character in the j^{th} position of the suffix is a_j. The same goes for the prefixes.

The Ambiguity Class: it stores all the possible tags assigned to a word during the training phase. It is represented in the pattern with a $|Tpos|$ vector of bits where the j^{th} bit is 1 iff t_j was assigned at least once to the word.

All these features computed for each word in the window, are merged into a single vector associated to the central word of the window. In the training phase, those vectors are provided to a set of SVM, one for each possible tag. The desired output of the j^{th} SVM is 1 iff t_j is the tag to be assigned to the word and 0 otherwise. When tagging a not previously seen sentence, the pattern is processed by all the SVMs. The one that predict a 1 with maximum confidence determines the tag to give to the central word of the window.

3 Results

We trained and tested the model using different configurations of parameters. Subsequently, we selected the 4 model reporting the higher scores on the validation test. The 4 models reporting the highest scores shared some basic parameters configuration:

- uses of a sliding window with a 5-words size;
- do not to consider Ambiguity Class as features;
- do not to consider prefixes of words;
- considering the 3-characters and the 4-characters suffixes for each word in the window.

The first model was derived using the basic configuration described above, obtaining an accuracy of 92,2% on sentences of the validation set. The second model uses the basic configuration with the inclusion of the 2-characters suffixes.

It reached an accuracy of 92,31%. The third configuration does not consider all the feature vector for the words in the window preceding the observed one, but only the tag previously assigned by the system. This model reported an accuracy of 92,51%. Finally the model reporting the highest score was obtained using a filling mechanism in the suffix analisys. It reached an accuracy of 92,89% on the validation set.

4 Discussion About Results

As described in the previous section, some features are not useful to improve the PoS tagging performances for Italian sentences. For example, the word-prefixes do not give any information about the POS of an italian word and their use negatively affects the learning process.

Despite the distribution of errors is quite uniform, we observed that the most frequent error is confusing nouns and adjectives. This can be explained considering that in a sentence often nouns and adjectives occupy more or less the same position and they are commonly located near a verb. As it is built, the system, taking into account the tag of words that surround the central, is led to commit this type of errors. As for verbs instead, it was possible to reduce the number of occurrences tagged incorrectly introducing into the model a greater number of suffixes using for example a suffix-gazetter.

References

1. Gimenez, J., Marquez, L.: Fast and accurate part-of-speech tagging: The svm approach revisited. In: Proceedings of RANLP, pp. 158–165 (2003)
2. Joachims, T.: Making large-scale support vector machine learning practical. In: Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, MA (1998)