A Light Sliding-Window Part-of-Speech Tagger for the Apertium Free/Open-Source Machine Translation Platform

Gang Chen, Mikel L. Forcada

Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, China
Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Spain
pkuchengang@gmail.com, mlf@dlsi.ua.es

Abstract

This paper describes a free/open-source implementation of the light sliding-window (LSW) part-of-speech tagger for the Apertium free/open-source machine translation platform. Firstly, the mechanism and training process of the tagger are reviewed, and a new method for incorporating linguistic rules is proposed. Secondly, experiments are conducted to compare the performances of the tagger under different window settings, with or without Apertium-style "forbid" rules, with or without Constraint Grammar, and also with respect to the traditional HMM tagger in Apertium.

Keywords: part-of-speech tagging, light sliding-window, machine translation, free/open-source

1. Introduction

Apertium[1] is a shallow-transfer rule-based free/open-source machine translation platform. This paper reports a free/open-source implementation of the light sliding window (LSW) PoS tagger (Sánchez-Villamil et al., 2005), and compares its performance with that of the original first-order HMM tagger in Apertium (Tyers et al., 2010; Sheikh and Sánchez-Martínez, 2009; Cutting et al., 1992). Section 2 reviews the mechanism of the LSW tagger and proposes a method to improve its tagging accuracy by incorporating linguistic rules, Section 3 shows the experimental results and discusses them, and finally, in Section 4, the paper ends with some conclusions and future plans.

2. Methods

The main difference between the LSW and HMM PoS taggers is that the LSW PoS tagger makes local decisions about the PoS tag of each word which are based on the ambiguity class (set of PoS tags) of words in a fixed-length context around the problem word, while HMM makes this decision by efficiently considering all possible disambiguations of all words in the sentence, by using a probabilistic model based on a multiplicative chain of transition and emission probabilities. In terms of model complexity, LSW is simpler than HMM, while, on the other hand, the number of parameters of LSW could be larger than that of HMM, which may have a crucial influence on the tagging performance as training material may not be sufficient to estimate them adequately.

The LSW tagger is an improved version of the sliding window (SW) PoS tagger (Sánchez-Villamil et al., 2004), and the main goal of the LSW tagger is to reduce the parameters of a SW tagger, by using approximations for the parameter estimation, without a significant loss in accuracy. Therefore, we briefly describe the SW tagger first, and then the LSW tagger.

2.1. The SW tagger

2.1.1. Overview

Let $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_{|\Gamma|}\}$ be the tag set, and $W = \{w_1, w_2, \ldots\}$ be the words to be tagged. A partition of W is established so that $w_i \equiv w_j$ if and only if both are assigned the same subset of tags, where each class of the partition is called an ambiguity class. Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$ be the collection of ambiguity classes, where each σ_i is an ambiguity class. Let $T : \Sigma \to 2^\Gamma$ be the function returning the collection $T(\sigma)$ of PoS tags for an ambiguity class σ.

The PoS tagging problem may be formulated as follows: given a text $w[1]w[2]\ldots w[L] \in W^+$, each word $w[t]$ is assigned (using a lexicon and a morphological analyzer) an ambiguity class $\sigma[t] \in \Sigma$ to obtain the ambiguously tagged text $\sigma[1]\sigma[2]\ldots\sigma[t] \in \Sigma^+$; the task of a PoS tagger is to obtain a tag sequence $\gamma[1]\gamma[2]\ldots\gamma[t] \in \Gamma^+$ as correct as possible, that is, the one that maximizes the probability of that tag sequence given the word sequence:

$$\gamma^*[1]\ldots\gamma^*[L] = \operatorname*{argmax}_{\gamma[t]\in T(\gamma[t])} P(\gamma[1]\ldots\gamma[L]|\sigma[1]\ldots\sigma[L]) \tag{1}$$

The core idea of SW PoS tagging is to use the ambiguity classes of neighboring words to approximate the dependencies locally:

$$P(\gamma[1]\ldots\gamma[L]|\sigma[1]\ldots\sigma[L]) = \prod_{t=1}^{t=L} p(\gamma[t]|C_{(-)}\sigma[t]C_{(+)}) \tag{2}$$

where $t = 1\ldots L$, $C_{(-)}$ is the left context of length $N_{(-)}$ (e.g. if $N_{(-)} = 1$, then $C_{(-)} = \gamma[t-1]$), and $C_{(+)}$ is the left context of length $N_{(+)}$.

2.1.2. Unsupervised parameter estimation

Let $p(\gamma|C_{(-)}\sigma C_{(+)})$ be the probability of a tag γ appearing between the context $C_{(-)}$ and $C_{(+)}$. The most probable tag $\gamma^*[t]$ is selected as the one with the highest probability by the formula:

$$\gamma^*[t] = \operatorname*{argmax}_{\gamma\in T(\sigma[t])} p(\gamma|C_{(-)}\sigma C_{+)}) \tag{3}$$

[1] The Apertium machine translation engine, linguistic data for various language pairs, and documentation can be downloaded from `http://www.apertium.org`.

Estimating the parameters from a tagged corpus would be straightforward, but estimating from an untagged corpus requires an iterative process. Let $\tilde{n}_{C_{(-)}\gamma C_{(+)}}$ (a simpler and interchangeable representation for $p(\gamma|C_{(-)}\sigma C_{(+)})$) be the effective number of times (count) that γ appears between the context $C_{(-)}$ and $C_{(+)}$. Following the steps in (Sánchez-Villamil et al., 2004), we can estimate $\tilde{n}_{C_{(-)}\gamma C_{(+)}}$ iteratively by:

$$\tilde{n}^{[k]}_{C_{(-)}\gamma C_{(+)}} = \tilde{n}^{[k-1]}_{C_{(-)}\gamma C_{(+)}} \sum_{\sigma:\gamma\in T(\sigma)} n_{C_{(-)}\sigma C_{(+)}} \left(\sum_{\gamma'\in T(\sigma)} \tilde{n}^{[k-1]}_{C_{(-)}\gamma' C_{(+)}} \right)^{-1} \tag{4}$$

A recommended initial value could be obtained by assuming that all the tags γ in σ are equally probable.

2.2. The LSW tagger

2.2.1. Overview

The SW tagger tags a word by looking at the ambiguity classes of neighboring words, and has therefore a number of parameters in $O(|\Sigma|^{N_{(-)}+N_{(+)}}|\Gamma|)$. The LSW tagger (Sánchez-Villamil et al., 2005) tags a word by looking at the possible tags of neighboring words, and therefore it has a number of parameters in $O(|\Gamma|^{N_{(-)}+N_{(+)}+1})$. Usually the tag set size $|\Gamma|$ is significantly smaller than the combinational ambiguity class size $|\Sigma|$. In this way, the number parameters is effectively reduced.

The LSW approximates the best tag as follows:

$$\gamma^* = \operatorname*{argmax}_{\gamma\in T(\sigma[t])} \sum_{\substack{E_{(-)}\in T'(C_{(-)}[t])\\ E_{(+)}\in T'(C_{(+)}[t])}} p(E_{(-)}\gamma E_{(+)}|C_{(-)}[t]\gamma[t]C_{(+)}[t]) \tag{5}$$

where $T' : \Sigma^* \to 2^{\Gamma^*}$, an extension of T, returns the set of tag sequences for an ambiguity sequence; $E_{(-)}$ and $E_{(+)}$ are the left and right tag sequence respectively.

2.2.2. Unsupervised parameter estimation

Following a procedure similar to that for the SW tagger, we can derive an iterative process to train the LSW tagger.

$$\tilde{n}^{[k]}_{E_{(-)}\gamma E_{(+)}} = \tilde{n}^{[k-1]}_{E_{(-)}\gamma E_{(+)}} \sum_{\substack{\sigma:\gamma\in T(\sigma)\\ C_{(-)}:E_{(-)}\in T'(C_{(-)})\\ C_{(+)}:E_{(+)}\in T'(C_{(+)})}} n_{E_{(-)}\sigma E_{(+)}} \left(\sum_{\substack{\gamma'\in T(\sigma)\\ C_{(-)}:E_{(-)}\in T'(C_{(-)})\\ C_{(+)}:E_{(+)}\in T'(C_{(+)})}} \tilde{n}^{[k-1]}_{E_{(-)}\gamma' E_{(+)}} \right)^{-1} \tag{6}$$

where $\tilde{n}_{E_{(-)}\gamma E_{(+)}}$ is the effective number of times (count) that γ appears between the context of tags $E_{(-)}$ and $E_{(+)}$. Similarly to the initialization step in the SW tagger, a recommended initial value can be obtained by assuming that all the tag sequences $E_{(-)}\gamma E_{(+)}$ in the window $C_{(-)}\sigma C_{(+)}$ are equally probable.

Items	Spanish	Catalan	English
Words (train)	3 million	4 million	3 million
Amb. classes (train)	106	92	68
Words (test)	25,000	25,000	30,000
Amb. rate (test)	22.81%	31.13%	29.97%
Forbid rules	545	272	117
Enforce rules	15	25	41

Table 1: Major statistics for the training and test data.

2.3. LSW with forbid and enforce rules

There are forbid and enforce rules for sequences of two PoS tags in the current implementation of the Apertium PoS tagger. They were successfully applied in the original HMM tagger in Apertium, with a significant improvement in accuracy (Sheikh and Sánchez-Martínez, 2009), simply by making the corresponding transition probabilities equal to zero. The SW tagger could not make use of forbid and enforce rules because of the fact that it works with ambiguity classes, while on the other hand, the LSW tagger can easily incorporate them as it works directly with PoS tags

The rules can be introduced right after the initialization step. For a tag sequence in the parameter space, if any consecutive two tags match a forbid rule or fail to match an enforce rule, the underlying parameter $\tilde{n}_{E_{(-)}\gamma E_{(+)}}$ will be given a starting value of zero.

In this way, for an LSW tagger with rules, the initial value could be given as follows,

$$\tilde{n}^{[0]}_{E_{(-)}\gamma E_{(+)}} = \begin{cases} 0 & \text{if } E_{(-)}\gamma E_{(+)} \text{ is not valid,} \\ \Lambda & \text{otherwise} \end{cases} \tag{7}$$

where

$$\Lambda = \sum_{\substack{\sigma:\gamma\in T(\sigma)\\ C_{(-)}:E_{(-)}\in T'(C_{(-)})\\ C_{(+)}:E_{(+)}\in T'(C_{(+)})}} n_{E_{(-)}\sigma E_{(+)}} \frac{1}{|V'(C_{(-)}\sigma C_{(+)})|} \tag{8}$$

where, the validity of $E_{(-)}\gamma E_{(+)}$ is determined by forbid and enforce rules, and the function V' returns the collection of valid (enforced or not forbidden) tag sequences contained in the ambiguity class sequence $C_{(-)}\sigma C_{(+)}$.

3. Experiments

3.1. Training data and test set

The experiments are conducted on three languages: Spanish (`apertium-en-es-0.8.0`), Catalan (`apertium-es-ca-1.1.0`), and English (`apertium-en-es-0.8.0`). We obtain the training data for Spanish and English by sampling text from the Europarl corpus (Koehn, 2005), and for Catalan by sampling text from the Catalan Wikipedia. The statistics on the training data and test data are shown in Table 1. Test data for Catalan and Spanish come from `apertium-es-ca-1.1.0`. It is worth noting that the English test set has been built by mapping the results form the TnT (Brants, 2000) tagger as an approximation.

3.2. The LSW tagger vs. the SW tagger

We firstly study whether there is a difference between the LSW tagger and the SW tagger, keeping all other settings the same. Then we study whether rules can help improve the accuracy for the LSW tagger. “Accuracy” in the graph refers to the tagging precision of a tagger on the hand-tagged test set. Figure 1 shows that rules help significantly for improving accuracy, and that the SW tagger behaves similarly to the LSW tagger without rules, which is consistent with the conclusion in (Sánchez-Villamil et al., 2005).

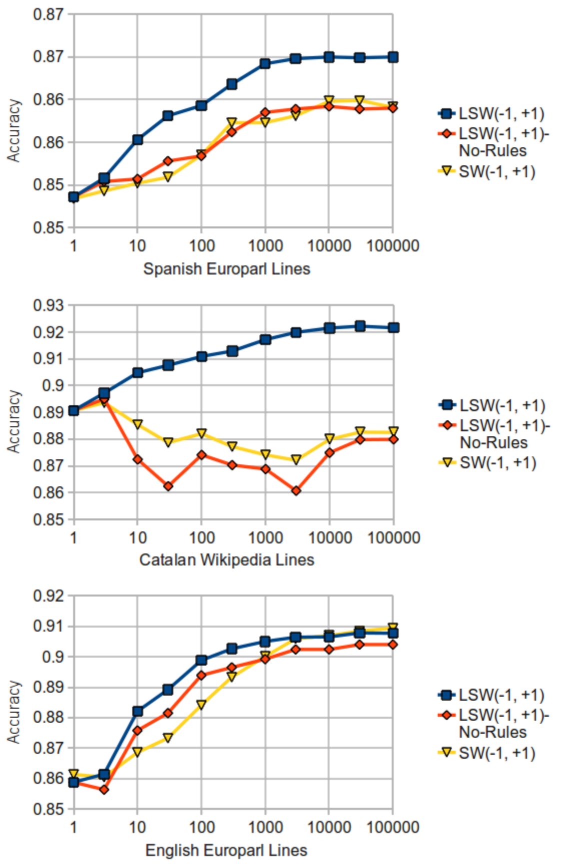

Figure 1: Performance evaluation for (1) the LSW(-1, +1) tagger, (2) the LSW(-1, +1) tagger without rules, denoted as LSW(-1, +1)-No-Rules, and (3) the SW(-1, +1) tagger, all on Spanish, Catalan, and English.

3.3. Different window settings for the LSW tagger

We study the performances of the LSW tagger with different window settings, and of the HMM tagger, on the three languages, as shown in Figure 2. We can see that the HMM tagger performs best among all the taggers, especially when there is enough training data. However, when training data is limited, the LSW taggers learn faster (need less words to learn) and more stably than the HMM tagger.

Among all the LSW taggers, the LSW(-1, +1), i.e. left context 1 and right context 1, performs best. When there are enough training data, the performances of the HMM tagger and the LSW(-1, +1) tagger are quite close.

Note that under some window settings, the performances of the LSW taggers even decrease as more training lines were added, e.g. LSW(-1) and LSW(-2, -1) for Spanish and Catalan. This is an unexpected phenomenon, and the reason for it would require further investigation.

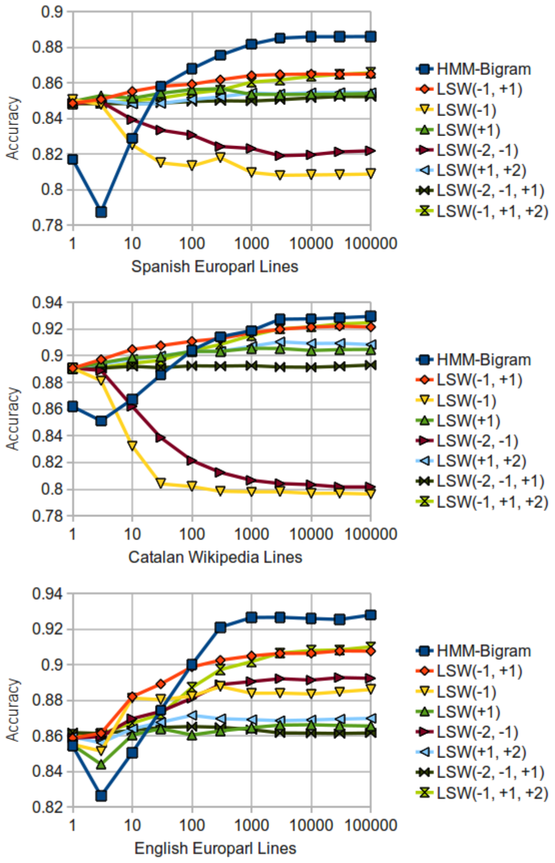

Figure 2: Different window settings and their performance, tested on Spanish, Catalan, and English.

3.4. Using Constraint Grammar rules to support the HMM and LSW

We also tested whether the use of Constraint Grammar (CG) rules helps to improve the accuracy obtained by both HMM and LSW taggers, along the lines suggested in (Hulden and Francom, 2012). For that, we used the CG rules already present in Apertium packages `apertium-eo-es-0.8.2` for Spanish and `apertium-eo-ca-0.8.2` for Catalan respectively (a CG module is integrated in many Apertium language pairs). Figure 3 shows that CG helps almost in all settings. It is also shown that CG rules help the two taggers in different situations: for the HMM tagger, the positive contribution of CG rules is larger when training data is limited than when training data is relatively enough; while for the LSW tagger, the trend is almost the opposite, that CG rules contribute even more when training data is relatively enough. Note that the logical approach would be to use CG rules both for reducing ambiguity for the training corpus (denoted as **cgTrain** in Figure 3) and for reducing ambiguity right after morphological analyzer and before the PoS tagger (denoted as **cgTag** in Figure 3); the results are however almost indistinguishable from those obtained applying CG in either step.

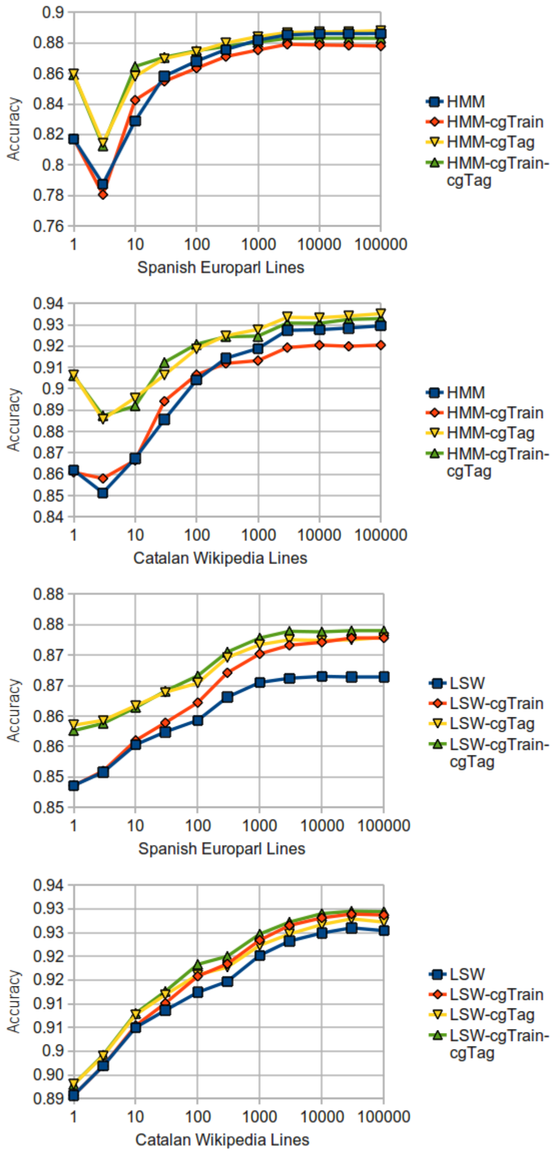

Figure 3: Performance evaluation for HMM and LSW with and without CG.

4. Discussion and future work

We reviewed the mechanism and unsupervised parameter estimation methods for both the SW and LSW taggers. Compared with previous work (Sánchez-Villamil et al., 2004; Sánchez-Villamil et al., 2005), firstly, we proposed a method for incorporating the forbid and enforce rules already used for HMM taggers in Apertium into the LSW tagger; and secondly, the implementation is the first time that the LSW tagger is integrated into a real machine translation system (Apertium), and at the same time, its code is free/open-source.

We also conducted experiments to compare the performances of the LSW tagger with different settings, and with respect to the original HMM tagger. Firstly, the HMM tagger performs slightly better than the LSW(-1, +1) tagger when there is enough training data, while the LSW(-1, +1) tagger learns faster and is more stable when training data is limited. Secondly, the LSW(-1, +1) tagger performs best among all the other window settings, and better than the SW(-1, +1) tagger, which behaves similarly with LSW(-1, +1)-No-Rules. Thirdly, we have found that the use of CG rule sets already existing in some Apertium taggers helps significantly to improve accuracy based both on the HMM and LSW taggers, and that for the HMM tagger CG rules help more when training data is limited, while for the LSW tagger CG rules help more when training data is relatively enough.

The reason why the performance of the LSW tagger under some window settings worsens as more training lines are added also requires more efforts to study. Source code is available through the Apertium Subversion repository[2] under a free/open-source license.

Acknowledgements: Support from Google Summer of Code (summer scholarship for Gang Chen) and from the Spanish Ministry of Economy and Competitiveness through grant TIN2012-32615 are gratefully acknowledged. The authors also thank Francis M. Tyers and Jim O'Regan for useful comments.

5. References

T. Brants. 2000. TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231.

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140.

M. Hulden and J. Francom. 2012. Boosting statistical tagger accuracy with simple rule-based grammars. In N. Calzolari, K. Choukri, T. Declerck, M. Ugur Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, editors, *LREC*, pages 2114–2117. European Language Resources Association (ELRA).

P. Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

E. Sánchez-Villamil, M. L. Forcada, and R. C. Carrasco. 2004. Unsupervised training of a finite-state sliding-window part-of-speech tagger. In *Advances in Natural Language Processing*, pages 454–463. Springer.

E. Sánchez-Villamil, M. L. Forcada, and R. C. Carrasco. 2005. Parameter reduction in unsupervisedly trained sliding-window part-of-speech taggers. In *Proceedings of Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September, 2005.

Z. M. A. W. Sheikh and F. Sánchez-Martínez. 2009. Parameter reduction in unsupervisedly trained sliding-window part-of-speech taggers. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 67–74. Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos.

F. M. Tyers, F. Sánchez-Martínez, S. Ortiz-Rojas, and M. L. Forcada. 2010. Free/open-source resources in the Aper-

[2] `https://svn.code.sf.net/p/apertium/svn/branches/apertium-swpost`

tium platform for machine translation research and development. *The Prague Bulletin of Mathematical Linguistics*, 93:67–76.