Unsupervised Learning of a Rule-based Spanish Part of Speech Tagger

Chinatsu Aone and **Kevin Hausman**
Systems Research and Applications Corporation (SRA)
4300 Fair Lakes Court
Fairfax, VA 22033
aonec@sra.com, hausman@sra.com

Abstract

This paper describes a Spanish Part-of-Speech (POS) tagger which applies and extends Brill's algorithm for *unsupervised learning* of *rule-based* taggers (Brill, 1995). First, we discuss our general approach including extensions we made to the algorithm in order to handle unknown words and parameterize learning and tagging options. Next, we report and analyze our experimental results using different parameters. Then, we describe our "hybrid" approach which was necessary in order to overcome a fundamental limitation in Brill's original algorithm. Finally, we compare our tagger with Hidden Markov Model (HMM)-based taggers.

1 Introduction

We have developed a Spanish Part-of-Speech (POS) Tagger which applies and extends Brill's algorithm for unsupervised learning (Brill, 1995) to create a set of rules that reduce the ambiguity of POS tags on words. We have chosen an *unsupervised learning* algorithm because it does not require a large POS-tagged training corpus. Since there was no POS-tagged Spanish corpus available to us and since creating a large hand-tagged corpus is both costly and prone to inconsistency, the decision was also a practical one. We have decided to develop a *rule-based* tagger because such a tagger learns a set of declarative rules and also because we wanted to compare it with Hidden Markov Model (HMM)-based taggers.

We extended Brill's algorithm in several ways. First, we extended it to handle unknown words in the training and test texts. Second, we parameterized learning and tagging options. Finally, we experimented with a "hybrid" solution, where we used a very small number of hand-disambiguated texts during training to overcome a fundamental limitation in the learning algorithm.

2 Components

Our Spanish POS tagger consists of three components: the Initial State Annotator, the Learner, and the Rule Tagger, each of which is described below.

2.1 Initial State Annotator

This component is used to assign all possible POS tags to a given Spanish word. It consists of lexicon lookup, morphological analysis, and unknown word handling. The Spanish POS tag set used in this work consists of the following tags: ADJ, ADV, BE (form of *ser* or *estar*), CLOCK-TIME, COLON, COMMA, CONJ, DATE, DET, HAVE (form of *haber*) , HYPHEN, LETTER, LPAREN, MODEL, MULTIPLIER, N, NUMBR, P, PERIOD, PREFIX, PRO, PROPN, QUES-MARK, QUOTE, ROMAN, RPAREN, SEMI-COLON, SLASH, SUBCONJ, SUFFIX, THERE (*hacer* used in "there" constructions), WHDET (*cuál libro*), WHNP (*qué*), WHPP (*dónde*), and V (See Table 3).

2.1.1 Lexicon Lookup and Morphological Analysis

Unlike Brill's English tagger experiment described in (Brill, 1995), no large POS-tagged Spanish corpus was available to us from which a large lexicon can be derived. As a result, we decided to parse the on-line Collins Spanish-English Dictionary[1], and derived a large lexicon from it (about 45,000 entries). We used only the open-class entries from this lexicon, and then augmented it with irregular verb forms and a number of closed-class words. Our morphological analyzer uses a set of rewrite rules to strip off and/or modify word endings to find root forms of words.

2.1.2 Unknown Word Handling

Since the lexicon and morphological analysis will not cover every single word that can appear in a text, an attempt is made at this stage to classify unknown words. Any word which did not get assigned one or more parts-of-speech in

[1] We have obtained a license to the dictionary.

the lookup/morphology phase is examined for certain traits often indicative of particular parts-of-speech. This task is similar to what was done by the *guessers* for the HMM-based French and German taggers (Chanod and Tapanainen, 1995; Feldweg, 1995).

For example, words ending in the letters "mente" are assigned the tag of ADV (adverb). Those words ending in "ando" or "endo" are assigned the tag V-CONTINUOUS-NIL (continuous form of the verb). Table 1 shows a list of unknown word handling rules.

Table 1: Unknown Word Handling Rules

Heuristics	POS tag
num > 1600 & < 2100	DATE
roman numeral 1-9	ROMAN
-and,-endo	V-CONTINUOUS-NIL
-ido,-ado,-ida,-ada	V-PERFECT-NIL
-er,-ir,-ar	V-NIL-NIL
-erse,-irse,-arse	V-NIL-NIL-CLITIC
-ción,-idad,-izaje	N
-mente	ADV
-able	ADJ
capitalized	PROPN

Performing these simple checks reduces the number of unknowns in our test set of 17,639 words from 737 (4.2%) to 158 (0.9%). The remaining unknowns are assigned a set of ambiguous open-class tags of N, V, ADJ, and ADV so that they can be disambiguated by the Learner.

2.2 Learner

The Learner takes as input ambiguously tagged texts produced by the Initial State Annotator, and tries to learn a set of rules that will reduce the ambiguity of the tags. Output is a file of rules in the following form:

$context = C : P_1 | \ldots | P_i | \ldots | P_n \rightarrow P_i$, where

context is one of PREVWORD, NEXTWORD, PREVTAG or NEXTTAG,[2]

C is a word or tag,

$P_1, \ldots, P_i, \ldots, P_n$ are the ambiguous parts-of-speech to be reduced,

P_i is the part-of-speech that replaces $P_1, \ldots, P_i, \ldots, P_n$.

Here are some examples taken from the actual learned rules:

- NEXTWORD = DE : P|N → N
- PREVWORD = EN : DET|ADV → DET
- PREVTAG = DET : V|N → N
- NEXTTAG = SUBCONJ : BE|V → V

The Learner applies Brill's algorithm for unsupervised learning to try to reduce the ambiguity of the tags in the input corpus. The following steps are taken:

1. The Learner examines each ambiguously tagged word and creates a set of contexts for the word. Two of these contexts will be PREVWORD and NEXTWORD. The remainder consist of PREVTAG and NEXTTAG contexts as required by the tag(s) on the preceding and following words. For example, if the word preceding the ambiguously tagged word is ambiguously tagged with two tags, then the Learner must generate two PREVTAG contexts.
2. An attempt is made to find unambiguously tagged words in the corpus that are tagged with one and only one of the tags on the ambiguously tagged word. For example, if the word in question has both N and V tags, then the Learner would search for words with only an N tag or only a V tag.
3. If such a word is found, the contexts of that word are examined to determine if there is an overlap between them and the contexts generated for the ambiguously tagged word. One issue for this determination is how much ambiguity should be tolerated in the context of the unambiguously tagged word. For example, if one of the possible contexts is PREVTAG=N and the word preceding the unambiguously tagged word has both N and V tags, should the context apply? To permit various approaches to be tried, we extended the Learner to accept a parameter (i.e., *freedom*) that determines how much ambiguity will be accepted on the context words for the context to match.
4. If a context matches for this unambiguously tagged word, the count of unambiguously tagged words with the particular part of speech occurring in that context is incremented.
5. After the entire corpus is examined, each of these possible reduction rules (of the form "Change the tag of a word from χ to Y in the context C where Y $\in \chi$") is ranked according to the following. First, for each tag Z $\in \chi$, Z $\neq$ Y, the Learner computes:

   $\frac{freq(Y)}{freq(Z)} * incontext(Z, C)$, where

   freq(Y)= number of occurrences of words unambiguously tagged with Y,

   freq(Z)= number of occurrences of words unambiguously tagged with Z,

   incontext(Z, C)= number of times a word unambiguously tagged with Z occurs in context C.

[2] PREVWORD = previous word, PREVTAG = previous tag.

The tag Z that gives the highest score from this formula is saved as R. Then the score for a particular transformation is

$$incontext(Y, C) - \frac{freq(Y)}{freq(R)} * incontext(R, C)$$

6. If the highest-ranked transformation is not positive, the Learner is done. Otherwise, the highest-ranked transformation is appended to the list of transformations learned. The Learner then searches this list for the transformation that will result in the most reduction of ambiguity (which will always be the latest rule learned) and applies it. This process continues until no further reduction of ambiguity is possible. Here, we also extended the Learner to accept a different parameter (i.e., *l-tagfreedom*) that determines how much ambiguity will be accepted on a word that is used for context during *ambiguity reduction*, that is, when the Learner has found a rule and is applying it to the training text. Note that specifying too small a value for this parameter can cause the Learner to go into an endless loop, as restricting the valid contexts may have the effect of nullifying the just-learned rule.
7. The Learner then returns to step 1 to begin the process again.

2.3 Rule Tagger

This component reads tagged texts produced by the Initial State Annotator and rules produced by the Learner and applies the learned rules to the tagged texts to reduce the ambiguity of the tags.

We extended the Rule Tagger to have two possible modes of operation (i.e., *best-rule-first* and *learned-sequence* modes controlled by the *seq* parameter) for using the learned rules to reduce ambiguity:

1. The Rule Tagger can use an algorithm similar to that used in step 7 of the Learner. Each possible reduction rule is examined against the text to determine which rule results in the greatest reduction of ambiguity.
2. The Rule Tagger can use a sequential application of the learned rules in the order that the rules were learned. After each rule has been applied in sequence, all of the rules preceding it are re-applied to take advantage of ambiguity reductions made by the latest rule applied.

The Rule Tagger allows one to specify, as in the Learner, how much ambiguity will be tolerated for a context to match. For example, one can be very restrictive and require that a tag context (e.g., PREVTAG=N) match only an unambiguously tagged word (in this case, a word with only an N tag). This parameter (i.e., *r-tagfreedom*) specifies the maximum ambiguity allowed on a context word for a context tag to match: 1 requires that the context word be unambiguously tagged, 2 requires that there be no more than two tags on the word, and so on.

3 Experiments and Results

For training and testing of the tagger, we have randomly picked articles from a large (274MB) "El Norte" Mexican newspaper corpus, and separated them into the training and test sets. The test set (17,639 words) was tagged manually for comparison against the system-tagged texts. For training, we partitioned the development set into several different-sized sets in order to see the effects of training corpus sizes. The breakdown can be found in Table 2.

Table 2: Ambiguously tagged Training sets

Set	Words
Tiny	1322 words
Small	3066 words
Medium	5591 words
Full	12795 words

If one randomly picks one of the possible tags on each word in the test set, the accuracy is 78.0% (78.6% with the simple verb tag set). The average POS ambiguity per word is 1.52 (1.49) including punctuation tags and 1.58 (1.56) excluding punctuation tags. For comparison, the accuracy of Brill's unsupervised English tagger was 95.1% using 120,000-word Penn Treebank texts. His initial state tagging accuracy was 90.7%, which is considerably higher than our Spanish case (78.6%).

3.1 Effect of Tag Set

Our first set of experiments tests the effect of the POS tag complexity. We used both the simple verb tag set (5 tags) and the complex verb tag set (42 tags), which is shown in Table 3, where * can be either 1SG, 2SG, 3SG, 1PL, 2PL, or 3PL. In the case of simple verb tag set, tense, person and number information is discarded, leaving only a "V" tag and the lower four tags in the table.

The scores with the simple verb tag set for different sizes of training sets are found in Table 4, and those with the complex verb tag set in Table 5. For these two experiments, the Learner was set to have a tight restriction on using context for learning (i.e, the *freedom* parameter was set to 1) and a loose restriction on context for applying the learned rules (i.e., *l-tagfreedom* 10). The Rule Tagger was given a moderately-tight restriction on using context for reduction rule application (i.e., *r-tagfreedom* 2).

In general, the scores are slightly higher using the simple verb tag set over the complex verb

Table 3: Complex Verb Tag Set

V-CONDITIONAL-*
V-FUTURE-*
V-IMPERFECT-*
V-IMPERFECT-SUBJUNCTIVE-RA-*
V-IMPERFECT-SUBJUNCTIVE-SE-*
V-PRESENT-*
V-PRESENT-SUBJUNCTIVE-*
V-PRETERIT-*
V-NIL-NIL
V-CONTINUOUS-NIL
V-PERFECT-NIL
V-NIL-NIL-CLITIC

Table 4: Ambiguously tagged texts, Simple Verbs

Set	# of rules learned	Score
Tiny	131	82.5%
Small	211	91.5%
Medium	287	91.8%
Full	434	83.0%
(none)	0	78.6%

tag set (91.8% vs. 90.3% for the "Medium" corpus). This behavior is most likely due to the fact that some verb tense/person/number combinations cannot easily be distinguished from context, so the Learner was unable to find a rule that would disambiguate them.

As can be seen from the tables, performance increased as the size of the learning set increased up to the "Medium" set, where the score levelled off. With very small learning sets, the system was unable to find sufficient examples of phenomena to produce reduction rules with good coverage.

One surprising data point in the simple verb tag set experiments was the "Full" score, which dropped almost 9% from the "Medium" score. After analyzing the results more closely, it was found that the Learner had learned a very specific rule regarding the reduction of preposition/subordinate-conjunction combinations late in the learning process. The learned rule was:

PREVTAG = N : P|SUBCONJ → SUBCONJ

Table 5: Ambiguously tagged texts, Complex Verbs

Set	# of rules learned	Score
Tiny	125	81.7%
Small	212	89.6%
Medium	323	90.3%
Full	564	90.2%
(none)	0	78.0%

This rule was learned late in the learning process when most P/SUBCONJ pairs had already been reduced. However, as one can see from the context of the rule, it will apply in a large number of cases in a text. The Rule Tagger notes this and applies the rule early, thus incorrectly changing many P/SUBCONJ pairs to SUBCONJ and reducing the accuracy of the tagging. Since this phenomenon never occurred in any of the other learning runs, one can see that the learning process can be heavily influenced by the choice of input texts.

3.2 Effect of Rule Application Parameters

The next tests performed involved using rules generated above and changing parameters to the Rule Tagger to see how the scores would be influenced. In the following test, we used the simple verb tag set rules but varied the *r-tagfreedom* parameter and the *seq* parameter. The results can be found in Table 6.

Table 6: Ambiguously tagged texts, Simple Verbs

Set	R-Tag-freedom	Score (best-rule-first)	Score (learned-sequence)
Tiny	1	82.7%	80.2%
	2	82.5%	80.6%
	3	82.1%	80.5%
	4	81.9%	80.5%
Small	1	90.1%	89.8%
	2	91.5%	89.9%
	3	91.5%	89.9%
	4	91.5%	89.9%
Medium	1	90.5%	90.6%
	2	91.8%	90.5%
	3	91.8%	90.5%
	4	91.8%	90.5%
Full	1	82.4%	79.8%
	2	83.0%	80.0%
	3	81.7%	80.0%
	4	81.5%	80.0%

Although the variations are slight, the best value for the *r-tagfreedom* parameter seems to be at an ambiguity level of 2. It seems that the strategy of reducing the ambiguity as quickly as possible (*best-rule-first*) is better than following the ordering of the rules by the Learner. This may well be due to the fact that the ordering of the rules as produced by the Learner is dependent on the training texts. Since the test set was a different set of texts, the ordering of the rules was not as applicable to them as to the training texts, and so the tagging performance suffered.

3.3 Effect of Hand-tagged Texts

After examining the results from the above experiments, we realized that some of the closed-class words in Spanish are almost always ambiguous (e.g., prepositions are usually ambiguous between PREP and SUBCONJ, and determiners between DET and PRO). This means that the Learner will *never* learn a rule to disambiguate these *closed-class* cases because there will rarely be unambiguous contexts in the training texts tagged by the Initial State Annotator. That is, unlike open-class words, we will not find new unambiguous closed-class words in texts precisely because there is only a *closed* set of them. Thus, we decided to introduce a small number of hand-tagged texts into the training set given to the Learner. Since the hand-tagged texts have "correct" examples of various phenomena, the Learner should be able to find good examples in them to learn from.

For our tests, we defined four sets of hand-tagged texts that we added to the "Small" (3066 words) set of ambiguously tagged texts. The breakdown is in Table 7.

Table 7: Hand-tagged Training sets

Set	Words
Small	218 words
Medium	588 words
Large	906 words
Full	1791 words

Again, the Learner was set to have a tight restriction on using context for learning (*freedom* 1) and a loose restriction on context for applying the learned rules (*tagfreedom* 10). The Rule Tagger was given a moderately-tight restriction on using context for reduction rule application (*freedom* 2). The *best-rule-first* mode of the Rule Tagger was used.

The results, as shown in Table 8, are slightly better than when using only ambiguously tagged texts. It is interesting to note that the higher accuracy was achieved with fewer rules. In fact, all experiments resulted in learning a little over 200 rules.

Table 8: Ambiguous/Unambiguous Texts, Simple Verbs

Set	# of rules learned	Score
Small	210	91.2%
Medium	211	92.1%
Large	205	92.1%
Full	202	92.1%
(none)	211	91.5%

In addition to the experiments above, we wanted to know if the introduction of hand-tagged texts into the "Full" ambiguously tagged set would improve its rather low score (cf. Table 4). We performed an experiment using simple verb tags, the "Full" ambiguously tagged texts, and the "Full" hand-tagged texts. The results were 422 rules learned with a score of 92.1%, which tied with the "Small" ambiguously tagged set for achieving the highest accuracy of all of the learning/tagging runs, a full 13.5% higher than using no learning.

4 Problems and Possible Improvements

Although our Spanish POS tagger performed reasonably well, achieving an improvement of 13.5% in accuracy over randomly picking tags, there were several problems that prevented the system from reaching an even higher score.

4.1 Learning Problem

As discussed in Section 3.3, ambiguous closed-class words (e.g., prepositions, determiners, etc.) cannot be reduced when there are no unambiguous examples of them in the training texts. This is prevalent in Spanish, where most prepositions can also be subordinate conjunctions, determiners can be pronouns, etc. A few hand-tagged texts are *required* to learn good rules for reducing the ambiguity on these words. It is possible, however, that such texts can be disambiguated only for their always ambiguous closed-class words but not unambiguous closed-class words or open-class words. Such an experiment similar to *selective sampling* discussed in Dagan and Engelson (Dagan and Engelson, 1995) would be useful in the future because, if it is true, it will reduce the cost of manual tagging considerably.

4.2 Lexicon Problem

Problems that became apparent as we ran more tests were the incompleteness and mistakes in the lexicon. While the lexicon, derived from the Collins Spanish-English dictionary, was quite rich in words, its tag set did not always match the tag definitions we employed. For example, our tag set distinguishes proper nouns (PROPN) and nouns (N), whereas the Collins dictionary marked both as nouns (N). We have added our existing proper name lists to the lexicon to partially solve this problem, but the lists are currently limited to location names and people's first names.

We also found several mistakes in the Collins definitions (e.g., several adverbs ending "-mente" were classified adjectives). Although we fixed these mistakes as we noticed them, it is difficult to know how many such errors still remain in the lexicon.

It turned out that the *incompleteness* of the lexicon was another fundamental problem to Brill's unsupervised learning algorithm. That is, when the lexicon does not list all the possible tags for a word, the tagger is very likely to make a mistake. This is because the learner is trained to reduce the ambiguity of possible tags of a word (say N, V, ADJ tags), but if the lexicon lists only a subset of the possible tags (say N and V tags), the system will *never* learn to assign an ADJ tag even when the word is used as an adjective.

This type of problem was observed frequently when words are ambiguous between proper nouns and some other parts-of-speech such as "Flores (ADJ/PROPN)," "Lozano (ADJ/PROPN)," "van (V/PROPN)"[3], "Serra (V/PROPN)," etc. because not all the proper nouns are in the lexicon.

The problems described above did not occur in Brill's experiments because he derived the lexicon from a POS-tagged corpus and used the untagged version of the same corpus for training and testing. Thus, he used an "optimal" lexicon which contains *all* the words with *only* parts-of-speech which appeared in the corpus. In addition, in such a corpus, rarely used POS tags of a word are less likely to occur, and words are less likely to be ambiguous. Thus, in a sense, his "unsupervised learning" experiments did take advantage of a large POS-tagged corpus.

5 Related Works

It is very difficult to compare performances between taggers when accuracy depends on quality of corpora and lexicons, and maybe on characteristics of languages. But in this section, we compare our tagger with Hidden Markov Model-based taggers.

A more widely used algorithm for unsupervised learning of a POS tagger is Hidden Markov Model (HMM). Cutting *et al.* (Cutting et al., 1992) and Melialdo (Merialdo, 1994) used HMM to learn English POS taggers while Chanod and Tapanainen (Chanod and Tapanainen, 1995), Feldweg (Feldweg, 1995), and León and Serrano (León and Serrano, 1995) ported the Xerox tagger (Cutting et al., 1992) to French, German, and Spanish respectively. One of the drawbacks of an HMM-based approach is that laborious manual tuning of symbol and transition biases is necessary to achieve high accuracy. Without tuned biases, the German Xerox tagger achieved 85.89% while the French Xerox tagger achieved 87% accuracy. After one man-month of tuning biases, the accuracy of the French tagger increased to 96.8%. One could derive such biases from a corpus, as discussed in (Merialdo, 1994), but it unfortunately requires a tagged corpus.

The best accuracy of the Spanish Xerox tagger was 91.51% for the reduced tag set (174 tags) with the base accuracy (i.e. no training) of 88.98% while the best accuracy of our tagger is currently 92.1% for the simple tag set (39 tags) with the base accuracy of 78.6%. The lower base accuracy in our experiment is probably due to the large number of entries in the Collins dictionary.

6 Summary

Our Spanish Part of Speech Tagger is a successful implementation and extension of Brill's unsupervised learning algorithm that reduces the ambiguity of part-of-speech tags on words in Spanish texts.

The system requires few, if any, hand-tagged texts to bootstrap itself. Rather, it merely requires a Spanish lexicon and morphological analyzer that can tag words with all their possible parts-of-speech. Given that the system performs at approximately 92% accuracy even with the aforementioned problems and with the inclusion of unknown words, we would expect that this system could achieve better results, approaching those of similar English-language POS taggers, when these problems are rectified.

References

Eric Brill. 1995. Unupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the 3rd Workshop on Very Large Corpora*.

Jean-Pierre Chanod and Pasi Tapanainen. 1995. Tagging French – Comparing statistical and a constraint-based method. In *Proceedings of the EACL-95*.

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*.

Ido Dagan and Sean P. Engelson. 1995. Selective Sampling in Natural Language Learning. In *Proceedings of the IJCAI Workshop on New Approach to Learning for Natural Language Processing*.

Helmut Feldweg. 1995. Implementation and Evaluation of a German HMM for POS Disambiguation. In *Proceedings of the EACL SIGDAT Workshop*.

Fernando Sánchez León and Amalio F. Nieto Serrano. 1995. Development of a spanish version of the xerox tagger. In *Proceedings of the XI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN '95)*.

Bernard Merialdo. 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2).

[3] It can be a part of a last name as in "van Mahler", but also is an inflected form of "ir".