Boosting Statistical Tagger Accuracy with Simple Rule-Based Grammars

Mans Hulden, Jerid Francom

Ikerbasque (Basque Foundation for Science), Wake Forest University
mans.hulden@email.arizona.edu, francojc@wfu.edu

Abstract

We report on several experiments on combining a rule-based tagger and a trigram tagger for Spanish. The results show that one can boost the accuracy of the best performing n-gram taggers by quickly developing a rough rule-based grammar to complement the statistically induced one and then combining the output of the two. The specific method of combination is crucial for achieving good results. The method provides particularly large gains in accuracy when only a small amount of tagged data is available for training a HMM, as may be the case for lesser-resourced and minority languages.

Keywords: part-of-speech tagging, constraint grammar, hybrid POS tagging, HMM taggers, Spanish

1. Introduction

Combining the tagging acumen of a rule-based and a statistical tagger in order to improve accuracy is an idea that is old, and sometimes also good. Experiments with such tagger combinations have been reported for various languages (Tapanainen and Voutilainen, 1994; Oflazer and Tür, 1996; Ezeiza et al., 1998; Hajič et al., 2001). Most documented hybrid tagging systems seem to improve upon using purely statistical methods. In more recent work, however, the advantage seems to be diminishing: while Tapanainen (1994) reports a reduction in the error rate of a statistical tagger for English by 4.92%, Spoustová et al. (2007), in an experiment with Czech, reduced the error rate of the output of a statistical tagger by 0.56%. The utility of rule-based grammars, therefore, appears to be declining as statistical methods improve. Granted, the differences in results may very well depend much on the language, the tagset, the training corpus and the particular method of combining two taggers.

A more important problem than declining utility is that previous efforts to combine the two methods have mostly mixed statistical taggers with very large handwritten grammars—grammars that may take years to develop and reach maturity. For example, (Tapanainen and Voutilainen, 1994) and (Spoustová et al., 2007) have both used Constraint Grammars (CGs) made up of thousands of rules. This raises the question of the cost/gain ratio of combining rule-based grammars with statistical ones.

The intuition behind the current work is that a large part of the rules in a detailed knowledge-based grammar tend to overlap in their action with the generalizations induced by a statistical grammar. Therefore, it would seem possible to develop a rudimentary rule-based grammar specifically designed to complement the statistical one, and thus to cheaply provide gains in tagging accuracy.

We have evaluated this possibility here through a number of experiments that involve training a statistical HMM tagger and combining it with a quickly developed, very rudimentary constraint grammar designed to produce high recall at the cost of precision. In other words, the goal of the rule-based grammar has been only to remove such tagging possibilities that are obviously (to the grammar writer) impossible, leaving other ambiguities unresolved.

More generally, the endeavor presented here corresponds to an old observation in machine learning—that multiple classifiers can be profitably combined, assuming they make complementary errors. In this particular case, we sacrifice precision to boost recall of the second classifier, with the goal of producing a complementary set of errors between the two. This enables us to merge them so that together they produce a higher accuracy than either could produce individually.

2. Overview

In order to evaluate the possibility of quickly boosting the performance of a statistical tagger, we have run various experiments with combining a small rule-based grammar (developed in a matter of days) that uses the Constraint Grammar formalism (Karlsson, 1990), and a run-of-the-mill Hidden Markov Model (HMM) tagger. The fundamental idea is to train the HMM tagger on a corpus, and then, while applying it, letting its options be limited to only the possibilities dictated by the rule-based grammar.

3. The training data

As the training and evaluation data we have used the *3LB* subset of the Ancora Spanish corpus (Taulé et al., 2008). This is the part of the corpus that, according to the authors, has had a manual post-correction of its tags, as opposed to the rest which is only automatically tagged. It consists of 94,775 token/tag pairs. The original tagset uses 271 distinct tags from which we have produced a reduced tagset of 65 tags, according to a simplification scheme suggested in the tagging guidelines of FreeLing (Carreras et al., 2004). This simplified tagset in essence removes agreement information from some of the original, more fine-grained tags. For example, some verb forms that have number and gender information in them such as *cantado/cantados/cantada/cantadas*, while originally represented as either **VMP00SM**, **VMP00PM**, **VMP00SF**, or **VMP00PF**, are all conflated into the general class **VMP**. For nouns, however, agreement information is retained. See the FreeLing tools (Carreras et al., 2004) for details.

The corpus was split randomly into 10 parts of roughly 9,500 tokens, one of which was set aside and marked as unavailable for the subsequent cross-validation testing task (though it was used for training). This was done in order to allow us to see a small part of the corpus for the purpose

of developing the Spanish CG; in this way, we could observe the tagging guidelines used and quickly modify the Constraint Grammar to bring it into line with the Ancora corpus without biasing the final results in favor of the CG analysis.

4. The CG grammar

The Constraint Grammar used for the task was originally developed as a high-recall grammar for preliminary tagging of an ongoing corpus project of Spanish.[1] The set of rules were developed during roughly 20 hours by the authors using a development corpus of 953 tokens (mainly using highly ambiguous sentences taken from a standard Spanish reference grammar (Bosque and Demonte, 1999)) for which it attained 100% recall. In total, the grammar consists of 148 rules, of which 113 are generic and 35 target specific word forms.

An example of a generic rule is one such as:

```
REMOVE (DET) IF (1C (VFIN));
```

which removes determiner readings if the following word has been deemed to unambiguously be a finite verb. An example of a specific rule is the rule

```
"<como>" SELECT (CS) IF (-1 (PUNC));
```

which selects the subordinating conjunction (over the relative pronoun) reading for the word *como* if it is preceded by punctuation.

As such, the grammar leaves many ambiguities unresolved. The overarching purpose is to remove only those ambiguities that one can "safely" eliminate.

4.1. The underlying morphological analyzer

The input to the constraint grammar is assumed to be morphologically analyzed, i.e. each word is tagged ambiguously with every possible morphological reading. To this end we use a finite-state morphological analyzer (SpanMorph) we have developed; the analyzer contains roughly 40,000 nouns, 19,000 adjectives, and 11,300 verbs. To extend its coverage, we merged it with the FreeLing Spanish dictionary (Carreras et al., 2004), producing a final finite-state transducer that recognizes roughly 3.2 million word forms and provides heuristic guesses for all other word forms based on suffixes and other morphological information.

5. The baseline HMM model

As our baseline model, we use a simple standard trigram tagger strategy that estimates a sequence of tags $t_1 \dots t_n$ from an input sequence of words $w_1 \dots w_n$ by maximizing

$$\prod_{i=1}^{n} p(t_i|t_{i-1}, t_{i-2})p(w_i|t_i) \quad (1)$$

The tag sequence counts learned during training are smoothed using Witten-Bell smoothing with backoff. Additionally, we build a separate letter model from the suffixes of the words up to length 6 to provide a model for $p(w_i|t_i)$ for unseen words during tagging. We build two different suffix models: one for words with an initial lowercase letter and downcased sentence-initial words, and another model for words with an initial uppercase letter (outside sentence-initial position). This is very similar to what the currently best-performing HMM taggers do (Brants, 2000; Halácsy et al., 2007), and as seen below, the baseline in fact gives a very similar result as plugging our training data and evaluation data into the freely available Hunpos tagger (Halácsy et al., 2007).

6. Experimental setup

We have conducted two basic experiments with combining the rule-based CG grammar and the baseline HMM tagger. All tests described below were run with 90% of the corpus used for training and 10% for testing, using 5-fold cross-validation. The testing data was subject to the constraint mentioned earlier that 10% of the entire corpus was marked off-limits for testing to avoid biasing the CG.

6.1. HMM and CG in parallel

In the first experiment, we trained our baseline HMM-tagger on the training data, but when tagging the testing set, we constrained the emissions of the tagger during the Viterbi search to choose only from those tags deemed possible by the constraint grammar in the context at hand. In other words, each emission probability $p(w_i|t_i)$ was set to 0 whenever the CG considered t_i not to be a possible tag for w_i for the context in which the probability was needed by the HMM tagger. In order to handle the case where the CG would allow a tag t_i for a word w_i, but where the combination was unseen during training of the HMM, we used add-δ smoothing for unseen tag-word pairs, setting $\delta = 0.2$.[2] The suffix-based unknown word-model used in the baseline HMM was suppressed when it was run in tandem with the CG, as the morphological analyzer was assumed to provide all the relevant possibilities for out-of-vocabulary items.

6.2. An HMM tagger with CG post-correction

Since delving into the innards of an HMM tagger, or programming one from scratch and modifying it to run in parallel with CG output, as described above, is a somewhat non-trivial task, we considered the possibility of taking an efficient off-the-shelf HMM tagger and combining it with our CG tagger in a sequential pipeline—an option available for anyone with access to two such pieces of software as the HMM tagger Hunpos (Halácsy et al., 2007) or TnT (Brants, 2000), and the CG tagger *vislcg3* (Bick, 2000).

To run this experiment, we used the freely available Hunpos tagger (Halácsy et al., 2007), trained it on the training data, and subsequently let it tag the evaluation data (always using its default options). We then tagged the same

[1] ACTIV-ES: a Spanish language corpus for linguistic and cultural comparisons between communities of the Hispanic world.

[2] One gets fairly similar results using a large range of δ. However, setting uniform emission probabilities (by setting $p(t_i|w_i) = 1/c$ where c is the number of possible tags, and then scaling to yield $p(w_i|t_i)$) for the words and tags actually worsens the result compared with only using the HMM. In other words, some care must be taken when constraining the HMM tagger with the constraint tagger.

	Morph	Morph+CG
Recall	99.41%	98.71%
Precision	58.10%	84.49%
Avg. Ambiguity	1.71	1.18

Table 1: *Evaluation of morphological analyzer, and CG component (148 rules).*

data with the constraint grammar, and subjected the output of the Hunpos tagger to a voting system: if the constraint tagger provided only one possible reading which disagreed with the Hunpos tagger, we used the CG tagger's choice in the final output. In the event that the CG had not completely disambiguated a certain token, we left the output of the Hunpos tagger untouched.

7. Results

7.1. Morphology and rule-based grammar only

As a preliminary, we evaluated the precision and recall of the morphological analyzer alone, and the morphological analyzer combined with the constraint grammar. These results are given in table 1. As can be seen, the recall of the CG is no longer perfect (as it was when developing it with the separate mini-corpus).

7.2. Combinations

The baseline for the experiment—running a pure HMM trigram tagger as described above—and the parallel combination of the HMM tagger and CG tagger are given in table 2. We also provide a list of the 10 most frequent mistaggings for the HMM+CG parallel system in table 4.

Parallel tagging		
	HMM	HMM+CG
Accuracy	96.32%	**97.67%**

Table 2: *Results for the trigram tagger, and the combined trigram and constraint grammar tagger.*

The second combination—running Hunpos and the constraint grammar tagger in series with a voting system—is given in table 3. We also give the result for training and tagging with only the Hunpos HMM tagger.

CG post-correction		
	Hunpos(HMM)	Hunpos+CG corrector
Accuracy	96.33%	97.46%

Table 3: *Results for running the freely available Hunpos tagger alone, and in series with the constraint grammar tagger.*

7.3. Learning curve

Additionally, we experimented with using varying amounts of training data for the HMM, from 1,000 tokens to 80,000 tokens, to produce a more holistic view of the learning curve when using only a HMM tagger versus using a HMM tagger in tandem with the CG tagger (with the parallel tagging approach). Figure 1 shows these comparative accuracies for the baseline HMM versus running the HMM and CG.

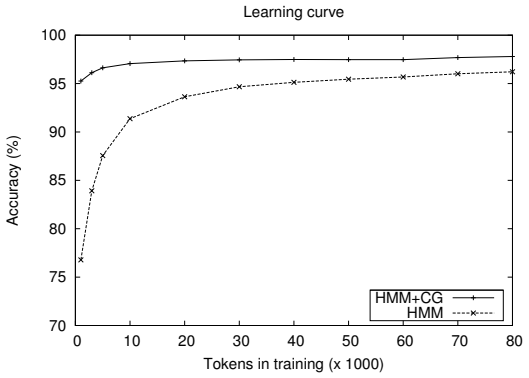

Figure 1: *Learning curve of HMM tagger vs. joint HMM+CG tagger on the Ancora 3LB Spanish corpus.*

Tagging errors		
Times	Correct	HMM+CG output
127	NC	AQ
127	AQ	NC
73	CS	PR
64	PR	CS
30	NC	NP
29	PP	P0
27	VMP	AQ
25	SP	CS
18	CC	RG
17	VMN	NC

Table 4: *Top 10 tagging errors by the HMM+CG joint tagger out of the total 1,104 errors made during 5-fold cross-validation.*

8. Discussion

As is evident from the results, combining the two taggers does indeed boost performance. This is especially evident the scarcer the training data is (see figure 1). In the second experiment where we combine an existing HMM and CG in a pipeline and a voting system, which is easier to implement, we see slightly less of an improvement (1.13%) in accuracy over a plain HMM tagger. With the embedded HMM and CG tagger we reach an accuracy of 97.67% — an improvement of 1.35% over the plain HMM tagger. Naturally, though, the former combination is far easier to implement than the latter as off-the shelf tools exist for both the HMM and CG parts and combining them in a pipeline is a straightforward task.

Looking at the tagging errors, foremost are the classical usual suspects: noun and adjective confusion. Many of the remaining errors appear to be a combination of a intrinsic error rate in the corpus—particularly prevalent are CS as PR tagging errors—something that is difficult to resolve without global information—as well as disagreement with the current authors and the Ancora corpus tagging guidelines. Because the Ancora corpus still contains traces of systematic mistaggings itself—probably due to it being initially tagger by a trigram tagger and then hand-corrected—qualitative evaluation becomes difficult as the error rate is already rather low. Based on a small sample of the corpus, we estimate that the error rate of the hand-corrected (but originally machine-tagged) Ancora corpus part used here is around $2-3\%$. We expect that with fewer errors in the corpus, the accuracy of the HMM+CG strategy would go up somewhat, but this is difficult to assess without resolving the systematic remaining tagging errors in Ancora. This is due to the fact that many of the CG rules are written with the explicit goal of resolving long-distance dependencies, and if tags that hinge on such dependencies are often incorrect in the corpus itself—a remnant of trigram tagging inaccuracy—the efforts of the rule-based grammar are nullified somewhat.

Below are a few example sentences that illustrate the top remaining errors, giving the correct tag first, followed by the incorrect tagging produced by the parallel HMM+CG system.

- De los 107.256 contratos de trabajo **indefinidos**_AQ/NC del pasado mayo ...
- ...nos va dando interferencias lingüísticas de blancos, **negros**_NC/AQ, y **cobrizos**_NC/AQ.
- De ahí su intento, **que**_PR/CS ya es una realidad ...
- ...siempre resulta más fácil y barato cultivar bacterias y virus **que**_CS/PR fabricar una bomba atómica.

These examples illustrate what seems to be a typical scenario with the remaining errors: many of them are difficult to resolve without semantic information. Particularly difficult is the disambiguation of *que* as a relative pronoun and *que* as a subordinating conjunction.[3] This is, of course, a very frequent token, and the relatively low accuracy of its disambiguation affects the overall accuracy to a large degree.

Another interesting result is the learning curve of a HMM together with a rudimentary CG disambiguator. As figure 1 shows, roughly 5,000 tagged words of training data for the HMM model when used together with the CG to yield a high level of accuracy ($> 96.4\%$), something not easily reached even by having more than a 100,000-word corpus when running only a HMM. This suggests that the HMM+CG approach could be particularly profitable in cases when there is little training data available, and producing tagged corpora would be cost-prohibitive.

[3]Except the last example where plausibly one could design a rule that precludes a pronoun reading of *que* in a context *noun-que-verb infinitival form*, which, despite the local nature of the generalization, is not modeled by the HMM tagger.

Acknowledgements

The first author has received partial research funding for this project from the European Commission's 7th Framework Program under grant agreement no. 238405 (CLARA). The second author has been funded under the National Endowment for the Humanities (NEH) grant number HD-51432-11.

9. References

E. Bick. 2000. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press.

Ignacio Bosque and Violeta Demonte. 1999. *Gramática descriptiva de la lengua española*. Espasa.

Thorsten Brants. 2000. TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied Natural Language Processing (ANLP-2000)*, pages 224–231.

Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th LREC*, volume 4.

N. Ezeiza, I. Alegria, J. M. Arriola, R. Urizar, and I. Aduriz. 1998. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 17th international conference on Computational linguistics—Volume 1*, pages 380–384.

J. Hajič, P. Krbec, P. Květoň, K. Oliva, and V. Petkevič. 2001. Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 268–275.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 209–212.

F. Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *Papers presented to the 13th International Conference on Computational Linguistics*, volume 3, pages 168–173, Helsinki, Finland.

Kemal Oflazer and Gökhan Tür. 1996. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 69–81.

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, pages 67–74.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately: don't guess if you know. In *Proceedings of the fourth conference on Applied Natural Language Processing (ANLP-1994)*, pages 47–52.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008)*.