Using Wiktionary to Build an Italian Part-of-Speech Tagger

Tom De Smedt[1], Fabio Marfia[2], Matteo Matteucci[2], and Walter Daelemans[1]

[1] CLiPS Computational Linguistics Research Group, University of Antwerp
tom@organisms.be walter.daelemans@uantwerpen.be
[2] DEIB Department of Electronics, Information and Bioeng., Politecnico di Milano
marfia@elet.polimi.it matteucci@elet.polimi.it

Abstract. While there has been a lot of progress in Natural Language Processing (NLP), many basic resources are still missing for many languages, including Italian, especially resources that are free for both research and commercial use. One of these basic resources is a Part-of-Speech tagger, a first processing step in many NLP applications. We describe a weakly-supervised, fast, free and reasonably accurate part-of-speech tagger for the Italian language, created by mining words and their part-of-speech tags from Wiktionary. We have integrated the tagger in Pattern, a freely available Python toolkit. We believe that our approach is general enough to be applied to other languages as well.

Keywords: natural language processing, part-of-speech tagging, Italian, Python

1 Introduction

A survey on Part-of-Speech (POS) taggers for the Italian language reveals only a limited number of documented resources. We can cite an Italian version of TreeTagger [1], an Italian model[3] for OpenNLP [2], TagPro [3], CORISTagger [4], the WaCky corpus [5], Tanl POS tagger [6], and ensemble-based taggers ([7] and [8]). Of these, TreeTagger, WaCky and OpenNLP are freely available tools. In this paper we present a new free POS tagger. We think it is a useful addition that can help the community advance the state-of-the art of Italian natural language processing tools. Furthermore, the described method for mining Wiktionary could be useful to other researchers to construct POS taggers for other languages.

The proposed POS tagger is part of Pattern. Pattern [9] is an open source Python toolkit for data mining, natural language processing, machine learning and network analysis, with a focus on user-friendliness (e.g., users with no expert background). Pattern contains POS taggers with language models for English [10], Spanish [11], German [12], French [13] and Dutch[4], trained using Brill's

[3] https://github.com/aparo/opennlp-italian-models
[4] http://cosmion.net/jeroen/software/brill_pos/

tagging algorithm (except for French). More robust methods have been around for some time, e.g., memory-based learning [14], averaged perceptron [15], and maximum entropy [16], but Brill's algorithm is a good candidate for Pattern because it produces small data files with fast performance and reasonably good accuracy.

Starting from a (manually) POS-tagged corpus of text, Brill's algorithm produces a lexicon of known words and their most frequent part-of-speech tag (aka a tag dictionary), along with a set of morphological rules for unknown words and contextual rules that update word tags according to the word's role in the sentence, considering the surrounding words. To our knowledge the only freely available corpus for Italian is WaCky, but, being produced with TreeTagger, it does not allow commercial use. Since Pattern is free for commercial purposes, we have resorted to constructing a lexicon by mining Wiktionary instead of training it with Brill's algorithm on (for example) WaCky. Wiktionary's GNU Free Documentation License (GFDL) includes a clause for commercial redistribution. It entails that our tagger can be used commercially; but when the data is modified, the new data must again be "forever free" under GFDL.

The paper proceeds as follows. In Sect. 2 we present the different steps of our data mining approach for extracting Italian morphological and grammatical information from Wiktionary, along with the steps for obtaining statistical information about word frequency from Wikipedia and newspaper sources. In Sect. 3 we evaluate the performance of our tagger on the WaCKy corpus. Sect. 4 gives an overview of related research. Finally, in Sect. 5 we present some conclusions and future work.

2 Method

In summary, our method consists of mining Wiktionary for words and word part-of-speech tags to populate a lexicon of known words (Sect. 2.1 and 2.2), mining Wikipedia for word frequency (Sect. 2.3), inferring morphological rules from word suffixes for unknown words (Sect. 2.5), and annotating a set of contextual rules (Sect. 2.6). All the algorithms described are freely available and can be downloaded from our blog post[5].

2.1 Mining Wiktionary for Part-of-Speech Tags

Wiktionary is an online "collaborative project to produce a free-content multilingual dictionary"[6]. The Italian section of Wiktionary lists thousands of Italian words manually annotated with part-of-speech tags by Wiktionary contributors. Since Wiktionary's content is free we can parse the HTML of the web pages to automatically populate a lexicon.

[5] http://www.clips.ua.ac.be/pages/using-wiktionary-to-build-an-italian-part-of-speech-tagger

[6] http://www.wiktionary.org/

We mined the Italian section of Wiktionary[7], retrieving approximately a 100,000 words, each of them mapped to a set of possible part-of-speech tags. Wiktionary uses abbreviations for parts-of-speech, such as *n* for nouns, *v* for verbs, *adj* for adjectives or *n v* for words that can be either nouns or verbs. We mapped the abbreviations to the Penn Treebank II tagset [20], which is the default tagset for all taggers in Pattern. Since Penn Treebank tags are not equally well-suited to all languages (e.g., Romance languages), Pattern can also yield universal tags [21], automatically converted from the Penn Treebank tags. Some examples of lexicon entries are:

- *di* → IN (preposition or subordinating conjunction)
- *la* → DT, PRP, NN (determiner, pronoun, noun)

Diacritics are taken into account, i.e., *di* is different from *dì*. The Italian section of Wiktionary does not contain punctuation marks however, so we added common punctuation marks (?!.:;,()[]+-*\) manually.

2.2 Mining Wiktionary for Word Inflections

In many languages, words inflect according to tense, mood, person, gender, number, and so on. In Italian, the plural form of the noun *affetto* (*affection*) is *affetti*, while the plural feminine form of the adjective *affetto* (*affected*) is *affette*. Unfortunately, many of the inflected word forms do not occur in the main index of the Italian section of Wiktionary. We employed a HTML crawler that follows the hyperlink for each word and retrieves all the inflected word forms from the linked detail page (e.g., conjugated verbs and plural adjectives according to the gender). The inflected word forms then inherit the part-of-speech tags from the base word form. We used simple regular expressions to disambiguate the set of possible part-of-speech tags, e.g., if the detail page mentions *affette* (plural adjective), we did not inherit the *n* tag of the word *affetto* for this particular word form.

Adding word inflections increases the size of the lexicon to about 160,000 words.

2.3 Mining Wikipedia for Texts

We wanted to reduce the file size, without impairing accuracy, by removing the "less important" words. We assessed a word's importance by counting how many times it occurs in popular texts. A large portion of omitted, low-frequency words can be tagged using morphological suffix rules (see Sect. 2.5).

We used Pattern to retrieve articles from the Italian Wikipedia with a spreading activation technique [22] that starts at the *Italia* article (i.e., one of the top[8] articles), then retrieves articles that link to *Italia*, and so on, until we reached

[7] http://en.wiktionary.org/wiki/Index:Italian
[8] http://stats.grok.se/it/top

1M words (=600 articles). We boosted the corpus with 1,500 recent news articles and news updates, for another 1M words. This biases our tagger to modern Italian language use.

We split sentences and words in the corpus, counted word occurrences, and ended up with about 115,000 unique words mapped to their frequency. For example, *di* occurs 70,000 times, *la* occurs 30,000 times and *indecifrabilmente* (*indecipherable*) just once. It follows that *indecifrabilmente* is an infrequent word that we can remove from our lexicon and replace with a morphological rule, without impairing accuracy:

-*mente* → RB (adverb).

Morphological rules are discussed further in Sect. 2.5.

2.4 Preprocessing a CSV File

We stored all of our data in a Comma Separated Values file (CSV). Each row contains a word form, the possible Penn Treebank part-of-speech tags, and the word count from Sect. 2.3 (Table 1).

Table 1. Top five most frequent words

Word form	Parts of speech	Word Count
di	IN	71,655
e	CC	44,934
il	DT	32,216
la	DT, PRP, NN	29,378
che	PRP, JJ, CC	26,998

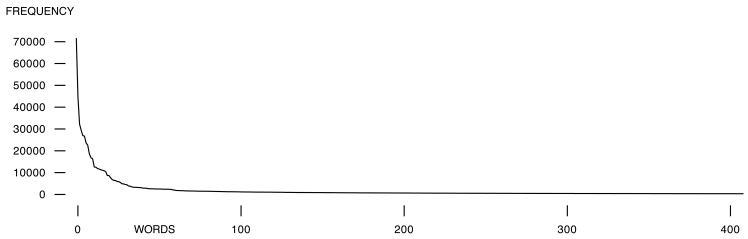

Fig. 1. Frequency distribution along words

The frequency distribution along words is shown in Fig. 1. It approximates Zipf's law: the most frequent word *di* appears nearly twice as much as the second most frequent word *e*, and so on. The top 10% most frequent covers 90% of popular Italian language use (according to our corpus). This implies that we

can remove part of "Zipf's long tail" (e.g., words that occur only once). If we have a lexicon that covers the top 10% and tag all unknown words as NN (the most frequent tag), we theoretically obtain a tagger that is about 90% accurate. We were able to improve the baseline by about 3% by determining appropriate morphological and contextual rules.

2.5 Morphological Rules Based on Word Suffixes

By default, the tagger will tag unknown words as NN. We can improve the tags of unknown words using morphological rules. One way to predict tags is to look at word suffixes. For example, English adverbs usually end in -*ly*. In Italian they end in -*mente*. In Table 2 we show a sample of frequent suffixes in our data, together with their frequency and the respective tag distribution.

We then automatically constructed a set of 180 suffix rules based on high coverage and high precision, with some manual supervision. For example:

-*mente* → RB

has a high coverage (nearly 3,000 known words in the lexicon) and a high precision (99% correct when applied to unknown words). In this case, we added the following rule to the Brill-formatted ruleset:

NN *mente* fhassuf 5 RB x.

In other words, this rule changes the NN tag of nouns that end in -*mente* to RB (adverb).

Table 2. Sample suffixes, with frequency and tag distribution

Suffix	**Frequency**	**Parts of speech**
-*mente*	2,969	99% RB, 0.5% JJ, 0.5% NN
-*zione*	2,501	99% NN, 0.5% JJ, 0.5% NNP
-*abile*	1,400	97% JJ, 2% NN, 0.5% RB, 0.5% NNP
-*mento*	1,375	99% NN, 0.5% VB, 0.5% JJ
-*atore*	1,28	84% NN, 16% JJ

2.6 Contextual Rules

Ambiguity occurs in most languages. For example in English, in *I can*, *you can* or *we can*, *can* is a verb. In *a can* and *the can* it is a noun. We could generalize this in two contextual rules:

- PRP + *can* → VB, and
- DT + *can* → NN.

We automatically constructed a set of 20 contextual rules for Italian derived from the WaCky corpus, using Brill's algorithm. Brill's algorithm takes a tagged corpus, extracts chunks of successive words and tags, and iteratively selects those chunks that increase tagging accuracy. We then proceeded to update and expand this set by hand to 45 rules.

This is a weak step in our approach, since it relies on a previously tagged corpus, which may not exist for other languages. A fully automatic approach would be to look for chunks of words that we know are unambiguous (i.e., one possible tag) and then bootstrap iteratively.

3 Evaluation

We evaluated the tagger (=Wiktionary lexicon + Wikipedia suffix rules + assorted contextual rules) on tagged sentences from the WaCky corpus (1M words). WaCky uses the Tanl tagset, which we mapped to Penn Treebank tags for comparison. Some fine-grained information is lost in the conversion. For example, the Tanl tagset differentiates between demonstrative determiners (DD) and indefinite determiners (DI), which we both mapped to Penn Treebank's DT (determiner). We ignored non-sentence-break punctuation marks; their tags differ between Tanl and Penn Treebank but they are unambiguous. We achieve an overall accuracy of 92.9% (=lexicon 85.8%, morphological rules +6.6%, contextual rules +0.5%).

Accuracy is defined as the percentage of tagged words in WaCky that is tagged identically by our tagger (i.e., for n words, if for word i WaCky says NN and our tag tagger says NN = $+1/n$ accuracy). Table 3 shows an overview of overlap between the tagger and WaCky for the most frequent tags.

Table 3. Breakdown of accuracy for frequent tags

NN	IN	VB	DT	JJ	RB	PRP	CC
(270,000)	(170,000)	(100,000)	(90,000)	(80,000)	(35,000)	(30,000)	(30,000)
94.8%	95.9%	88.5%	90.0%	84.6%	88.6%	82.1%	96.8%

For comparison, we also tested the Italian Perceptron model for OpenNLP and the Italian TreeTagger (Achim Stein's parameter files) against the same WaCky test set. With OpenNLP, we obtain 97.1% accuracy. With TreeTagger, we obtain 83.6% accuracy. This is because TreeTagger does not recognize some proper nouns (NNP) that occur in WaCky and tags some capitalized determiners (DT) such as *La* and *L'* as NN. Both issues would not be hard to address.

We note that our evaluation setup has a potential contamination problem: we used WaCky to obtain a base set of contextual rules (Sect. 2.6), and later on we used the same data for testing. We aim to test against other corpora as they become (freely) available.

4 Related Research

Related work on weakly supervised taggers using Wiktionary has been done by Täckström, Das, Petrov, McDonald and Nivre [17] using Conditional Random Fields (CRFs); by Li, Graça and Taskar [18] using Hidden Markov Models (HMMs); and by Ding [19] for Chinese. CRF and HMM are statistical machine learning methods that have been successfully applied to natural language processing tasks such as POS tagging.

Täckström et al. and Li et al. both discuss how Wiktionary can be used to construct POS taggers for different languages using bilingual word alignment, but their approaches to infer tags from the available resources differ. We differ from those works in having inferred contextual rules both manually and from a tagged Italian corpus (discussed in Sect. 2.6).

Ding used Wiktionary with Chinese-English word alignment to construct a Chinese POS tagger, and improved the performance of its model with a manually annotated corpus. Hidden Markov Models are used to infer tags.

5 Future Work

We have constructed a simple POS tagger for Italian by mining Wiktionary, Wikipedia and WaCky with weak supervision, contributing to the growing body of work that employs Wiktionary as a useful resource of lexical and semantic information. Our method should require limited effort to be adapted to other languages. It would be sufficient to direct our HTML crawler to another language section on Wiktionary, with small adjustments to the source code. Wiktionary supports, up to now, about 30 languages.

As discussed in Sect. 4, other related research uses different techniques (i.e., HMM), often with better results. In future research we want to compare our approach with those taggers and verify what is the gap between our and other methods.

Finally, Pattern has functionality for sentiment analysis for English, French and Dutch, based on polarity scores for adjectives (see [9]). We are now using our Italian tagger to detect frequent adjectives in product reviews, annotating these adjectives, and expanding Pattern with sentiment analysis for the Italian language.

References

1. Schmid, H. (1994, September). Probabilistic part-of-speech tagging using decision trees. In Proceedings of international conference on new methods in language processing (Vol. 12, pp. 44-49).
2. Morton, T., Kottmann, J., Baldridge, J., Bierner, G. (2005). Opennlp: A java-based nlp toolkit.
3. Pianta, E., Zanoli, R. (2007). TagPro: A system for Italian PoS tagging based on SVM. Intelligenza Artificiale, 4(2), 8-9.

4. Tamburini, F. (2009). PoS-tagging Italian texts with CORISTagger. In Proc of EVALITA 2009. AI*IA Workshop on Evaluation of NLP and Speech Tools for Italian.
5. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language resources and evaluation, 43(3), 209-226.
6. Attardi, G., Fuschetto, A., Tamberi, F., Simi, M., Vecchi, E. M. (2009). Experiments in tagger combination: arbitrating, guessing, correcting, suggesting. In Proc. of Workshop Evalita (p. 10).
7. Søgaard, A. (2009). Ensemble-based POS tagging of Italian. In The 11th Conference of the Italian Association for Artificial Intelligence, EVALITA. Reggio Emilia, Italy.
8. Dell'Orletta, F. (2009). Ensemble system for Part-of-Speech tagging. In Proceedings of EVALITA, 9.
9. De Smedt, T., Daelemans, W. (2012). Pattern for Python. The Journal of Machine Learning Research, 98888, 2063-2067.
10. Brill, E. (1992, February). A simple rule-based part of speech tagger. InProceedings of the workshop on Speech and Natural Language (pp. 112-116). Association for Computational Linguistics.
11. Reese, S., Boleda, G., Cuadros, M., Padró, L., Rigau, G. (2010). Wikicorpus: A word-sense disambiguated multilingual Wikipedia corpus.
12. Schneider, G., Volk, M. (1998). Adding manual constraints and lexical look-up to a Brill-tagger for German. In Proceedings of the ESSLLI-98 Workshop on Recent Advances in Corpus Annotation, Saarbrücken.
13. Sagot, B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In 7th international conference on Language Resources and Evaluation (LREC 2010).
14. Daelemans, W., Zavrel, J., Berck, P., Gillis, S. (1996, August). MBT: A memory-based part of speech tagger generator. In Proceedings of the Fourth Workshop on Very Large Corpora (pp. 14-27).
15. Collins, M. (2002, July). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 1-8). Association for Computational Linguistics.
16. Toutanova, K., Klein, D., Manning, C. D., Singer, Y. (2003, May). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 173-180). Association for Computational Linguistics.
17. Täckström, O., Das, D., Petrov, S., McDonald, R., Nivre, J. (2013). Token and type constraints for cross-lingual part-of-speech tagging. Transactions of the Association for Computational Linguistics, 1, 1-12.
18. Li, S., Graça, J. V., Taskar, B. (2012, July). Wiki-ly supervised part-of-speech tagging. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 1389-1398). Association for Computational Linguistics.
19. Ding, W. (2012). Weakly supervised part-of-speech tagging for chinese using label propagation.
20. Marcus, M. P., Marcinkiewicz, M. A., Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. Computational linguistics,19(2), 313-330.

21. Petrov, S., Das, D., McDonald, R. (2011). A universal part-of-speech tagset.arXiv preprint arXiv:1104.2086.
22. Collins, A. M., Loftus, E. F. (1975). A spreading-activation theory of semantic processing. Psychological review, 82(6), 407.