Mivoq Evalita 2016 PosTwITA tagger

Giulio Paci
Mivoq Srl
Padova - Italy
giulio.paci@mivoq.it

Abstract

English. The POS tagger developed by Mivoq to tag tweets according to PosTwITA task guidelines as defined at Evalita 2016 is presented. The system obtained third position with 92.7% of accuracy.

Italiano. *Si presenta il POS tagger sviluppato da Mivoq per etichettare i tweet secondo le linee guida del task PosTwITA, cos come definite per Evalita 2016. Il sistema ha ottenuto la terza posizione con un accuratezza del 92.7%.*

1 Introduction

Twitter messages (Tweets) are challenging for Natural Language Processing (NLP) due to the conversational nature of their content, the unconventional orthography and the 140 character limit of each tweet (Gimpel et al., 2011). Moreover tweets contain many elements that are not typical in conventional text, such as emoticons, hashtags, at-mentions, discourse markers, URL and emails.

Text-To-Speech (TTS) systems make large use of NLP technologies as part of input preprocessing, in order to cope with:

- homographs disambiguation: TTS systems may use POS tagging as a preliminary step to identify the correct pronunciation for those words that shares the same written form, but are pronounced differently. Many Italian homographs can be disambiguated using the POS tag (e.g., the string "ancora" has two possible pronunciations according to the fact that we are referring to the noun "anchor" or to the adverb "still"), although in some cases more information is needed;
- words expansion: as not all the text correspond to pronounceable words, TTS systems need to convert some strings into pronounceable words (e.g., numbers, units, acronyms, URL, ...). POS tags are useful to identify the function of a string and perform correct expansion (e.g., the string "1" can be expanded into "uno", "un" and "una", according to the POS tags of the surrounding strings);
- prosody prediction: prosody includes several aspects of speech, such as intonation, stress, tempo, rhythm and pauses, that are often not perfectly encoded by grammar or by choice of vocabulary, but still are important for the communication and should be correctly rendered by TTS systems. POS tags correlate with several prosodic aspects (e.g., content words are generally produced with more prominence than function words) and thus are useful for prosody prediction.

This work is the first attempt of the author to develop a POS tagger suitable for usage in a TTS system dealing with tweets.

2 Description of the system

The proposed system is the combination of several taggers and resources:

- Hunpos (Halácsy et al., 2007), an open-source reimplementation of TnT (Brants, 2000) HMM based tagger;
- Yamcha (Kudo and Matsumoto, 2003), an open-source Support Vector Machine (SVM) based tagger;
- CRFSuite (Okazaki, 2007), a Conditional Random Fields (CRF) based tagger;
- Evalita 2016 PosTwITA training data set;

- Evalita 2009 POS Tagging training data set (Attardi and Simi, 2009; Attardi et al., 2008; Zanchetta and Baroni, 2005), this corpus comprises 108,874 word forms divided into 3,719 sentences extracted from the online edition of the newspaper "La Repubblica" and annotated using the Tanl tag-set;

- ISTC pronunciation dictionary: originally developed for the Italian module of the Festival Text-To-Speech system (Cosi et al., 2001), has been later expanded by several contributors and currently includes pronunciations of 3,177,286 distinct word forms. POS tag information (using Tanl tag-set) has been added to each pronunciation for the purpose of pronunciation disambiguation; for this reason this information is reliable for all those words with multiple possible pronunciations, but many admissible tags may be missing for the other entries.

Six different taggers, corresponding to different combinations of these resources, have been tested in a 10-fold cross-validation scheme. Three taggers have been trained on the PosTwITA training data and thus can be used independently to solve the Evalita 2016 PosTwITA task. Two of them have been trained on Evalita 2009 Pos Tagging training data and can be used to solve that task instead. The sixth tagger combines the above taggers and is the system that has been proposed.

2.1 Hunpos

Hunpos has been used as a black box, without the use of an external morphological lexicon: an attempt have been made to use the ISTC pronunciation dictionary, but performance degraded. Hunpos has been trained on PosTwITA training data, where it obtained an average accuracy of 92.51%, and on Evalita 2009 Pos Tagging training data, where it obtained and average accuracy of 95.72%.

2.2 Yamcha

Yamcha allows the usage of arbitrary features and can be used to implement a wide range of taggers. Features combinations are implicitly expanded using a polynomial kernels and exploited by SVM (Kudo and Matsumoto, 2003).

Several feature sets have been tested, using the default parameters for Yamcha (i.e., only pair wise multi class method and second degree polynomial kernel have been used). Yamcha has been trained on PosTwITA training data and obtained an average accuracy of 95.41%.

2.2.1 Baseline

The baseline experiment with Yamcha consists in using features proposed by its author for English POS-tagging (Kudo, 2003 2005):

- the string to be annotated (i.e., the word);

- three Boolean flags set to true if: the string contains a digit, the string contains non alphanumeric characters, the first character of the string is an upper case character;

- the suffixes and prefixes of the string (with character length from 1 to 4, set to __nil__ if the original string is shorter than the suffix or the prefix length).

The default feature window has been used in this experiment (i.e., for each word form, features for previous two word forms and next two word forms are used, as long as the annotation results of the previous two word forms). Average accuracy is reported in table 1.

2.2.2 Twitter specific elements

A rule-based annotator for typical twitter elements (Prescott, 2012 2013) has been implemented:

- hashtags: an hash followed by a string composed by word characters only (the actual implementation allow some common symbols in the string, such as apostrophe, dots or &, thus matching author intention rather than proper twitter syntax);

- at-mentions: an optional dot, followed by an @ symbol, followed by a valid username (the actual implementation do not validate usernames and allows some common symbols in usernames);

- URLs (common mistakes are handled and matched in the implementation);

- emoticons: rules have been added to match both western (e.g., ":-)", ":-(", ...) and Asian (e.g., "^_^", "UwU", ...) style emoticons, to handle characters repetitions (e.g., ":-)))") and to match a subset of Unicode emoji. The rules have been tuned on a set of emoticons

described in Wikipedia (Wikipedia users, 2004 2016) and expanded according to the author's experience.

Although the accuracy improvement due to this feature was marginal (see table 1), it was present in all the tests and allowed almost perfect match of all Twitter specific elements, which is very important for words expansion.

2.2.3 Normalized string

Phonetic normalization has been proposed to deal with the many alternate spelling of words in English tweets (Gimpel et al., 2011). In this work a much simpler normalization is used, consisting in consecutive duplicated characters removal and converting to lower case. The following feature set has been tested:

- the string to be annotated (i.e., the word);
- three Boolean flags set to true if: the string contains a digit, the string contains non alphanumeric characters, the first character of the string is an upper case character;
- the suffixes and prefixes of the string (with character length from 1 to 3, set to __nil__ if the original string is shorter than the suffix or the prefix length);
- the prefixes and suffixes of the normalized string (with character length from 1 to 4 and 1 to 6 respectively).
- Twitter elements rule-based annotation.

In order to reduce the number of features, prefixes, suffixes and twitter annotations of the surrounding words has not been considered. The system achieved an average accuracy of 94.61%.

2.2.4 Dictionary tags

Finally 12 Boolean flags has been added, by performing a dictionary lookup using the normalized strings. Each flag corresponds to a PosTwITA tag (VERB_CLIT, VERB, INTJ, PROPN, NOUN, ADJ, ADP, ADP_A, SYM, ADV, DET, NUM) and is set to true if the ISTC dictionary contains a Tanl POS tag that can be mapped into it. By adding this feature the system achieved and average accuracy of 95.41%.

2.3 CRFSuite

The same feature sets used with Yamcha have been tested with CRFSuite, leading to very similar results, as shown in table 1. CRFSuite has been trained on both PosTwITA and on Evalita 2009 Pos Tagging training data sets, obtaining similar accuracy for both.

2.4 Tagger combination

The final system is a combination of five taggers based on Yamcha, by adding their output to the feature set. Tags associated to the surrounding tokens (3 previous and 3 next) are considered: using a larger window helped reducing errors with AUX and VERB tags. Results for individual taggers and the final system are shown in table 1. The system achieved an average accuracy of 95.97%. Implementing the same system using only the three taggers trained on PosTwITA data, lead to a very similar average accuracy of 95.74%, however the proposed system achieved better results in all the tests.

3 Results

	Hunpos	**Yamcha**	**CRFSuite**
	Evalita 2009 POS Tagging		
Hunpos	**95.72%**		
YN+T+D			**95.41%**
	Evalita 2016 PosTwITA		
Hunpos	**92.51%**		
YB		93.17%	93.02%
YB+T		93.30%	
YN+T		94.61%	94.17%
YN+T+D		**95.41%**	**95.31%**
MT		**95.97%**	

Table 1: 10-fold cross-validation average accuracy of a few configurations on both Evalita 2009 POS Tagging and Evalita 2016 PosTwITA training sets.

Table 1 reports average accuracy obtained in 10-fold cross-validation experiments on Evalita 2016 PosTwITA and Evalita 2009 POS Tagging data sets. Each column corresponds to a different tagger and each row corresponds to a different feature set, as described in section 2. YB is the baseline feature set described in section 2.2.1, YB+T is the baseline feature set with rule-based Twitter elements' annotation described in section 2.2.2, YN+T is the feature set described in section 2.2.3

	Hunpos	Yamcha	CRFSuite
Hunpos	**85.90%**		
	(86.43%)		
YB		88.95%	88.86%
		(89.75%)	(89.58%)
YB+T		88.91%	
		(89.72%)	
YN+T		90.10%	90.25%
		(91.01%)	(90.83%)
YN+T+D		**91.36%**	**92.06%**
		(92.27%)	(92.71%)
MT		**92.71%**	
		(93.74%)	

Table 2: blind test average accuracy of a few configurations.

and YN+T+D is the YN+T feature set with the addition of dictionary usage as described in section 2.2.4. MT is the final system described in section 2.4. Table 2 reports accuracy results for the same configurations on the PosTwITA test set. In this case results after manual correction of the test set are reported below the official results.

4 Discussion

Results in table 1 and table 2 shows that Yamcha and CRFSuite behave very similarly. By using YN+T+D feature set, CRFSuite achieves accuracy similar to that of Hunpos on the Evalita 2009 POS Tagging training set. With that feature set, performance of CRFSuite on both Evalita 2009 POS Tagging and Evalita 2016 PosTwITA training sets are very similar, suggesting the idea that YN+T+D feature set is quite stable and can be used successfully for both tasks. It would be interesting to include similar features in Hunpos in order to confirm the hypothesis.

Results on the Evalita 2016 PosTwITA test set shows a big accuracy loss, suggesting a mismatch between the training and the test sets. Manual correction of the test set, performed by the author, alleviated the differences, but still results are not comparable. Table 3 reports the 10 most frequent tokens in the PosTwITA training and test sets. The test set includes only function words and punctuation, but the most frequent word in the training set is the proper noun "Monti" and the word "governo" (government) is also among the most frequent tokens. Including at-mentions, hashtags and without considering the case, the word "monti"

Training set		*Test set*	
3362	.	124	,
2908	,	85	e
2315	**Monti**	82	.
2148	di	77	di
2109	:	66	che
1915	il	66	a
1652	e	64	”
1503	che	52	?
1499	a	50	:
1437	**governo**	49	...

Table 3: 10 most frequent tokens in PosTwITA training and test sets.

appears in 3460 tokens, making it the most frequent token in the data set and suggesting a very narrow topic. On the other hand the test set topic seems more general: the most frequent tokens are either words or punctuation marks and the first proper noun, "Italia" (Italy), appears at position 43. Given the topic mismatch, the tagger combination seems more stable than individual taggers.

The author goal was to investigate the possibility to implement a POS tagger suitable for reading tweets within a TTS system. Confusing NOUN and PROPN tags, and confusing ADJ, NOUN, AUX and VERB tags (in particular with nouns derived from adjectives or with adjectives derived from verbs) are among the most frequent errors. These errors do not typically affect the pronunciations. Hashtags, at-mentions and URL are correctly recognized with just one error, so that correct expansion of these elements can be performed. Several emoticons were wrongly annotated as punctuation, due to the limited set of Unicode emoji recognized by the rule-based annotation system and can be easily fixed by extend the match to the whole Unicode emoji set.

The difference in terms of accuracy between CRFSuite with YN+T+D feature set and the tagger combination, does not seem to justify the overhead of running multiple taggers; it would be interesting to train the taggers on a more general data set, eventually using the proposed tagger to bootstrap its annotation. Assuming that the pronunciation dictionary is already available in the TTS, the YN+T+D feature set described in section 2 seems appropriate for the POS tagging task for both tweets and more conventional text.

References

Giuseppe Attardi and Maria Simi. 2009. Overview of the evalita 2009 part-of-speech tagging task. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence*, Reggio Emilia, Italy.

Giuseppe Attardi et al. 2008. Tanl (text analytics and natural language processing). URL: `http://medialab.di.unipi.it/wiki/SemaWiki`.

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics. doi:10.3115/974147.974178.

Piero Cosi, Fabio Tesser, Roberto Gretter, Cinzia Avesani, and Michael W. Macon. 2001. Festival speaks italian! In *7th European Conference on Speech Communication and Technology*.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: `http://dl.acm.org/citation.cfm?id=2002736.2002747`.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212, Stroudsburg, PA, USA. Association for Computational Linguistics. URL: `http://dl.acm.org/citation.cfm?id=1557769.1557830`.

Taku Kudo and Yuji Matsumoto. 2003. Fast methods for kernel-based text analysis. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 24–31, Stroudsburg, PA, USA. Association for Computational Linguistics. doi:10.3115/1075096.1075100.

Taku Kudo. 2003-2005. Yamcha: Yet another multipurpose chunk annotator. URL: `http://chasen.org/~taku/software/yamcha/`.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). URL: `http://www.chokkan.org/software/crfsuite/`.

Adam Prescott. 2012-2013. twitter-format - syntax and conventions used in twitter statuses. URL: `http://aprescott.github.io/twitter-format/twitter-format.7`.

Wikipedia users. 2004-2016. Emoticon. In *Wikipedia*. URL: `https://it.wikipedia.org/wiki/Emoticon`.

Eros Zanchetta and Marco Baroni. 2005. Morph-it! a free corpus-based morphological resource for the italian language. In *PROCEEDINGS OF CORPUS LINGUISTICS*. URL: `http://dev.sslmit.unibo.it/linguistics/morph-it.php`.