Part-of-Speech Tagging for Code mixed English-Telugu Social media data

by

Kovida Nelakuditi, jittadivya.sai , Radhika Mamidi

in

17th International Conference on Intelligent Text Processing and Computational Linguistics

MEXICO.

Report No: IIIT/TR/2016/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2016

Part-of-Speech Tagging for Code mixed English-Telugu Social media data

Kovida Nelakuditi, Divya Sai Jitta, and Radhika Mamidi

Kohli Center on Intelligent Systems (KCIS)
International Institute of Information Technology, Hyderabad (IIIT Hyderabad)
Gachibowli, Hyderabad, Telangana 500032
{nelakuditi.kovida, jittadivya.sai}@research.iiit.ac.in,
radhika.mamidi@iiit.ac.in

Abstract. Part-of-Speech Tagging is a primary and an important step for many Natural Language Processing Applications. POS taggers have reported high accuracies on grammatically correct monolingual data. This paper reports work on annotating code mixed English-Telugu data collected from social media site Facebook and creating automatic POS Taggers for this corpus. POS tagging is considered as a classification problem and we use different classifiers like Linear SVMs, CRFs, Multinomial Bayes with different combinations of features which capture both context of the word and its internal structure. We also report our work on experimenting with combining monolingual POS taggers for POS tagging of this code mixed English-Telugu data.

Keywords: Code mixing, Social media data, Part-of-Speech tagging

1 Introduction

Code-mixing refers to the mixing of two or more languages or language varieties in speech. [1] This phenomenon is extended to writing as well. It is the embedding of various linguistic units such as affixes, words, phrases and clauses of one language in the other. [1]

India is a land of many languages. People on social media often use more than one language to express themselves. In this paper, we present our work on building a POS tagger for English-Telugu code mixed data collected from social media site Facebook. Telugu is the third most spoken language in the country according to 2001 census[1] and a good percentage of educated Telugu speaking people use English in their daily speech. It could be the use of borrowed words/phrases or code-mixing. We would not make any distinction between borrowing and mixing in this paper. We refer to both the phenomena as code-mixing and handle them alike. Telugu is an agglutinative language belonging to Dravidian language family, with 74 million native speakers. English is the language of instruction in most academic places and is also used for daily communication by

[1] http://www.mapsofindia.com/culture/indian-languages.html

both educated and uneducated population of India. Code-mixing happens at a pervasive rate in this setting. In this paper, we discuss the nature of such data, annotation of it and explore different ways of building a POS tagger for it. These methods include use of Machine Learning algorithms and combination of POS taggers of individual languages.

Mixing happens at many levels in English-Telugu code mixed data owing to the rich morphological nature of Telugu. [2]

- Full borrowing:
  - Phonological changes: Borrowed words from English or other languages are used in Telugu with a sound change. This is done to achieve nativization.
    Eg: zebra *jIbrA*[2]
  - Lexical borrowings: Words from English are used in Telugu phrases/sentences. (The inverse phenomenon, though happens, is very rare.)
    Examples:
    1. *nIku* help *cesAnu*.
       English: I helped you.
       In this sentence we can observe that the English word 'help' is used in a Telugu sentence.
    2. *velYlYi* trainu *eVkku*.
       English: go, get on the train.
       In this sentence, we can observe the borrowed word, train is used in the sentence with a suffix, '-u' added at the end. This kind of change happens over a wide range of borrowed English words.
- Partial borrowing:
  - Native suffixation: The words borrowed from English are changed according to the morphological nature of Telugu. Telugu is a postpositional language and agglutinative in nature with morphemes affixing to each other at the end of the root word. The roots can also be borrowed English words, resulting in words with Telugu suffixes and English roots.
    Examples:
    train – *trainu*
    1. trains – *trainlu*
       (*trainu*+*lu*, *'lu'* is the plural marker in Telugu)
    2. 'in train' – *trainlo*
       (*trainu*+*lo*, *'lo'* is the locative marker in Telugu)
    3. 'in trains' – *trainullo* (*trainu*+*'lu'*+*'lo'*)
    4. 'for train' – *trainuki* (*trainu*+*'ki'*, *'ki'* is the beneficiary marker in Telugu)
  - Agglutination in Complex Verbs: A complex verb is a multi-word compound with one noun and a verbal component. Due to agglutinative nature of Telugu, we have complex verbs with borrowed English words acting as nouns that are agglutinated to Telugu verbs.

[2] Telugu examples are written in italics and are in wx-format (sanskrit.inria.fr/DATA/wx.html).

Examples:

1. *luncayiMxi* – lunch+*ayiMxi*
   had lunch
2. *resultoVcciMxi* – result+*voVcciMxi*
   result is out

- Syntactic Changes: Grammar of one language influences the other.

1. Word order: English is a SVO language and Telugu is a SOV language. Due to influence of English, unnatural yet comprehensible Telugu constructions are seen in daily usage.
   Example:
   Telugu: *mA sabbulo uMxi nimma Sakwi*
   English: Our soap has the power of lemon
   In the Telugu utterence *'uMxi'* is the verb, *'mA sabbulo'* is the subject of the sentence and *'nimma SAkwi'*, the object. A more natural construction for the above utterance would be:
   Telugu: *mA sabbulo nimma Sakwi uMxi*
   This phenomenon is commonly observed in advertisements and dubbed movies because of the shortcomings in dubbing techniques.
2. Pro-drop: Telugu is a pro-drop language, whereas English is not. The data is observed to have English sentences with initial subject dropped.
   Example:
   Indian English: will go to the Market after a while.
   Native English: I will go to the Market after a while.

POS tagging is the process of marking up a word in a text as corresponding to a particular part of speech. POS tagging is an important step for many Natural Language Processing Applications like Machine Translation, Dialog systems, parsing etc. Code mixed data from social media is noisy in nature with non-standard spellings and creative style of writing by the users. This nature of social media data poses additional problems. Supervised POS tagging accuracies for English measured on the PennTree bank have converged to around 97.3%. [3] Supervised POS tagging for Telugu[3] has an accuracy of 76.01%. [4] As a part of this work we explore various methods for creating a POS tagger for code-mixed English-Telugu data.

2 Data Annotation

The data is annotated with language labels and POS tags for 6570 words and is shared for NLP Tools Contests: POS Tagging Code-Mixed Indian Social Media

[3] ltrc.iiit.ac.in/analyzer/telugu/shallow-parser-tel-3.0.fc8.tgz

Text @ ICON 2015. Further this data is annotated with language labels, normalized form of the word, POS tag of it and chunk level information for 10207 words. We use this data to perform our experiments. This data is annotated by two individual annotators at four levels: which are 1. Language of the word, 2. Correct form of the word i.e the citation form of the word as available in the dictionary, 3. The part-of-speech tag of the word, 4. Chunk information of the word. We have used the first three levels of the annotated corpus for the experiments described in the paper.

1. Language Identification: Language Identification is the process of identifying the language of a particular word. Every word is tagged with one of the labels: T, E, M, N and R.
   T class has words which belong to Telugu language.
   E class has words which belong to English language.
   N class has named entities.
   M class has words with English roots and Telugu morphological inflections. Examples :-
   (a) postlu (post+'lu') posts, here the plural marker, 'lu' from telugu.
   (b) supere (super + 'e') super+clitique 'e'
   R class has rest of the words which include words from other languages, URLs, symbols etc.
2. Transliteration/Normalisation: The correct form of the word, i.e the citation form of the word as available in the dictionary of the corresponding language, is provided along with the word. If the word belongs to T or M then Brahmi script or wx format[4] is used to represent the word, this is the form available in a Telugu dictionary. Brahmi script and wx format have a one to one character mapping. For words belonging to class E, the entries available in a dictionary of English are used.
3. Part-of-speech: Each word is annotated with its part of speech. We use universal POS tagset[5] to tag the data. It acts as a common tagset with 16 tags that can be used for all the languages.
4. Chunking: Chunking is the process of identifying and labeling different types of phrases such as Noun phrase, Verb phrase, Adjectival phrase etc in a sentence. The annotation is at word level (a word is a segment with a space on both sides) and we annotate each word as beginning (B) or intermediate of a chunk (I), along with type of the phrase. For example, the tag B-NP against a word indicates that that word is the starting word of the Noun phrase, and I-NP indicates that it is the intermediate word of Noun phrase. A phrase starts at the beginning tag('B-') and includes all words between (but not) the word with next beginning tag.
   Chunking part of the annotation is carried out as we plan to implement chunking in future on code mixed English-Telugu social media data. The experiments described in the paper are on developing a POS tagger and do not concern with the chunking.

[4] wx-format (sanskrit.inria.fr/DATA/wx.html)
[5] http://universaldependencies.org/docs/u/pos/

The annotation of an example sentence from the corpus, 'plz watch it nd share chusaka nenu cheppanavasarm le meere share chestharuuu' is shown in Table 1 and the statistics of the annotated corpus is described in Table 2.

Table 1. 5-level annotation for an example sentence

plz	E please	ADV	B-VP
watch	E watch	VERB	I-VP
it	E it	PRON	B-NP
nd	E and	CONJ	B-CP
share	E share	VERB	B-VP
chusaka	T cUsAka	VERB	B-VP
neenu	T nenu	PRON	B-NP
cheppanavasaram	T ceVppanavarasaraM	NOUN	B-NP
le	T lexu	VERB	B-VP
meere	T mere	PRON	B-NP
share	E share	NOUN	B-NP
cheestaru	T ceswAru	VERB	B-VP

Table 2. Statistics of the annotated corpus

#words	10207
#sentences	1335
Average length of a sentence	8
#words in class E	4515
#words in class T	4342
#words in class R	379
#words in class M	81
#words in class N	890

3 Related Work

POS taggers on monolingual data give an accuracy of about 97.3% for English [3] and 76.01% for Telugu [4]. They are often seen as sequence labeling problems and have used the context based information in the form of lexical and sub-lexical characteristics of neighboring words. But in code-mixed setting, the context information can be in a different language which makes the understanding difficult.

Work has been done on English-Hindi and English-Bengali data. [5] is a POS tagger developed for English-Hindi code mixed data, where the authors have used monolingual Hindi and English POS taggers and combined the output

from the two taggers. [6] describes tagger with combination of two monolingual POS taggers and also four Machine Learning algorithms on English-Hindi code mixed data. Typologically, Hindi is defined as inflectional where as Telugu is defined as agglutinative. [7] As a part of this work we have annotated English-Telugu data of 10,207 words and we have used three different Machine Learning algorithms to build the POS taggers and also experiment with combining POS taggers of individual languages.

Unknown words typically cause problems for POS tagging systems. Words from social media are noisy and fall under the category of OOV words while using existing systems. Normalization is an important and necessary step for proper working of existing POS taggers on this data. [8] developed a POS tagger for twitter data. No such work has been done for Telugu to date. We use CMU's Twitter POS tagger [8] for POS tagging of words identifed as belonging to English, this tagger has a normalization module in itself. [9] is a transliteration system built for Indian languages, we use this system for transliteration of Telugu words.

4 Approach

We perform two different kinds of experiments:

1. Machine learning based POS taggers
2. Combining POS taggers of individual languages

4.1 Machine learning based POS taggers

We use three kinds of Machine Learning algorithms for building the POS tagger viz, Support Vector Machines(SVM), Bayes classification(Bay), Conditional Random Fields(CRF) with different combinations and variations of the following features:

- lexical feature:
  - word(CW)
- sub-lexical features: This set of features include varying length of prefix, suffix, infix character strings derived from the current word and its neighboring words.
  - Prefix, suffix character strings(CPS): Telugu is a postpositional language which is mostly suffixing and English is a prepositional language, due to this reason having prefix, suffix character strings as features helps better understanding of the language and determining the POS tag of it.
  - Infix character strings(CI): Telugu is agglutinative, many bound morphemes can fuse to form a word, we include infix character strings to the feature set. This, in addition to suffix character strings can help in finding more number of bound morphemes.
  - Presence of postpositions(PSP): We check for the presence of Telugu postpositions at the end of the word and use it as a feature.

    - Prefix, suffix character strings of neighboring words(NPS)
- other features:
    - length of the word(WL)
    - neighboring words(N)

4.2 Combining POS taggers of individual languages

Solorio and Liu [10] have proposed to combine POS taggers of individual languages for the code-mixed data. We have used CMU's Twitter POS tagger [8] for English and POS tagger developed at LTRC, which is a part of the shallow parser tool [1] for Telugu.

The pipeline of this system is as follows:

1. Language Identification: Language Identification is the process of dividing words into one of the mentioned classes. A CRF model is implemented for this purpose with the following features.
    - lexical feature:
        - word
    - sub-lexical features:
        - Prefix, suffix character strings
        - Infix character strings
        - Presence of Post positions
        - prefix, Suffix character strings of neighboring words
    - other features:
        - length of the word
        - neighboring words
2. Transliteration/normalization: We use CMU pos tagger on English words (M) which reported an accuracy of 89.39% [8], it normalizes English words as a primary step. We use IRTRANS [9] for Telugu words (T) and mixed words (M). This tool is used to convert roman into Telugu script i.e Brahmi. However, no normalization for words of class M and T is possible at this stage due to lack of resources.
3. POS tagging: We use POS taggers built for monolingual data i.e we use both English and Telugu POS taggers. We give the words of class T and M to Telugu POS tagger and the words of E and R to English POS tagger. We don't process words belonging to class N as it is directly mapped to proper noun (PROPN). We later map the tagsets of English POS tagger and Telugu POS tagger to universal POS tagger.

5 Experiments and Results

We have used CRF++ tool [11] for implementing CRF, implemented Bayes classification and SVM using Scikit-learn [12].

[1] ltrc.iiit.ac.in/analyzer/telugu/shallow-parser-tel-3.0.fc8.tgz

5.1 Machine Learning based POS taggers:

The following table describes the accuracies obtained from three different Machine Learning algorithms namely, CRFs, SVMs and Bayes classification with different combinations of features. Accuracies are reported on three-fold and five-fold cross validation.

Table 3. Results of different feature templates on 3-fold cross validation.

Template	CPS	CI	NW	NPS	PSP	SVM	Bayes	CRF
1	3,2	3	0	0	yes	67.78	65.27	73.58
2	3,2,1	3	1	3	yes	65.69	61.46	75.75
3	3,2,1	3	2	3	yes	64.12	59.29	**76.22**
4	3,2,1	3	1	3,2	yes	60.17	51.96	75.66
5	3,2,1	3	1	2	yes	65.65	63.66	75.90

Table 4. Results of different feature templates on 5-fold cross validation.

Template	CPS	CI	NW	NPS	PSP	SVM	Bayes	CRF
1	3,2	3	0	0	yes	70.46	67.34	77.88
2	3,2,1	3	1	3	yes	68.25	63.51	80.98
3	3,2,1	3	2	3	yes	66.47	61.17	**81.40**
4	3,2,1	3	1	3,2	yes	63.10	53.46	81.05
5	3,2,1	3	1	2	yes	68.48	63.66	80.09

In both the above tables, the following conventions are followed:

- If CPS = k, k length suffix and prefix strings of the current word are taken as a feature.
- If CPS = k1, k2,.. then k1, k2,.. length suffix and prefix strings of the current word are taken as a feature.
- If NPS = k, k length suffix and prefix strings of neighboring words are taken as a feature.
- If NPS = k1, k2,.. then k1, k2,.. length suffix and prefix strings of neighboring words are taken as feature.
- If CI=k, k length infix string form the current word is taken as a feature.

We can observe that template3 gives more accuracy and CRF performs better than SVMs and Bayes classifier. In all the above experiments word length (WL) and current word (CW) are taken as features.

5.2 Combining POS taggers of individual languages:

Language Identification:
Using different combinations and variations of these features, various templates were experimented upon and the accuracies of three-fold cross validation on the dataset are reported in the following table:

Table 5. Results of different feature templates for Language Identification on 3-fold cross validation

Template	CPS	CI	WL	NPS	NW	PSP	CRF
1	3	3	yes	0	0	no	89.94
2	3,2,1	0	no	3	2	no	92.65
3	3,2,1	3	yes	3	2	yes	93.01
4	3,2,1	0	yes	3	1	yes	92.76
5	3,2	3	yes	3	1	yes	92.33
6	3	3	no	0	1	no	91.01
7	3,2,1	3	no	3	2	no	**93.08**

We can see that template 7 gives the best accuracy and we select this template for further use. After Language Identification, words recognized as belonging to T and M classes are transliterated into Brahmi script using IRTRANS [9]. After transliteration, words belonging to classes M and T are run on POS tagger of Telugu[3] and words belonging to class E and R are run on CMU's Twitter POS tagger for English. We finally map the tagset of Telugu POS tagger [13] and CMU's Twitter POS tagger [8] to universal POS tagset to find POS tags of all the words. It is to be noted that words belonging to class N are mapped to PROPN by this stage. The accuracy of the individual module is 58.66% as tested on the total dataset of 10207 words. The accuracy of the pipeline is 52.37% as tested on training data of 5840 words and testing data of 4367 words.

We observe that including infix strings of length three along with prefix/suffix strings and using the information of presence or absence of postposition in the feature set gives good accuracy as Telugu is a agglutinative language which has multiple morphemes fusing together and we are able to capture the internal structure of the word to some extent using these features. In addition to these, we get the contextual information from neighboring words and we also used features which describe their internal structure. The performance of combination of different language taggers is not so satisfactory compared to the Machine Learning approaches because of three main reasons:

1. Propagation of error - As this is a pipeline system, there is a definite error propagation from Language Identification and Transliteration/Normalisation stages.

[3] ltrc.iiit.ac.in/analyzer/telugu/shallow-parser-tel-3.0.fc8.tgz

2. Non-availability of normaliser for Telugu - Telugu words are complex in constructions. As no normaliser has been built for it to date, the POS tagger of Telugu fails in analysing many words.
3. Loss of context information - With words from different languages in the same sentence, not all the words within a sentence are fed to the POS tagger of a single language as is the case in taggers of monolingual data(in that case POS tagger use the contextual information to perform their best). Hence there is a loss of contextual information in combining POS taggers of individual languages.

6 Conclusion

The paper describes the first step at building shallow parser for English-Telugu code mixed data. Through this paper we present our efforts at attempting various statistical methods for POS tagging of code-mixed social media data. We also create a standard dataset for building supervised models of shallow parsing on this data which we consider as our immediate future work. POS tagging is a necessary and essential step for many NLP applications and the results that we have obtained are encouraging.

References

1. Ayeomoni, M.O.: Code-switching and code-mixing: Style of language use in childhood in yoruba speech community. Nordic Journal of African Studies **15**(1) (2006) 90–99
2. Kosaraju, P., Kesidi, S.R., Ainavolu, V.B.R., Kukkadapu, P.: Experiments on indian language dependency parsing. Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing (2010)
3. Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: Computational Linguistics and Intelligent Text Processing. Springer (2011) 171–189
4. Rao, D., Yarowsky, D.: Part of speech tagging and shallow parsing of indian languages. Shallow Parsing for South Asian Languages (2007) 17
5. Vyas, Y., Gella, S., Sharma, J., Bali, K., Choudhury, M.: Pos tagging of english-hindi code-mixed social media content. In: EMNLP. Volume 14. (2014) 974–979
6. Jamatia, A., Gambäck, B., Das, A.: Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. RECENT ADVANCES IN (2015) 239
7. Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Jha, G.N., et al.: A common parts-of-speech tagset framework for indian languages. In: In Proc. of LREC 2008, Citeseer (2008)
8. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: 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, Association for Computational Linguistics (2011) 42–47

9. Bhat, I.A., Mujadia, V., Tammewar, A., Bhat, R.A., Shrivastava, M.: Iiit-h system submission for fire2014 shared task on transliterated search. In: Proceedings of the Forum for Information Retrieval Evaluation, ACM (2014) 48–53
10. Solorio, T., Liu, Y.: Part-of-speech tagging for english-spanish code-switched text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2008) 1051–1060
11. Kudo, T.: Crf++: Yet another crf toolkit. Software available at http://crfpp. sourceforge. net (2005)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research **12** (2011) 2825–2830
13. Bharati, A., Sangal, R., Sharma, D.M., Bai, L.: Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. LTRC-TR31 (2006)