Part-of-Speech Annotation of English-Assamese code-mixed texts: Two Approaches

Ritesh Kumar
Department of Linguistics
K.M. Institute of Hindi and Linguistics
Dr. Bhimrao Ambedkar University, Agra
riteshkrjnu@gmail.com

Manas Jyoti Bora
Department of Linguistics
K.M. Institute of Hindi and Linguistics
Dr. Bhimrao Ambedkar University, Agra
manasjyotimj@gmail.com

Abstract

In this paper, we discuss the development of a part-of-speech tagger for English-Assamese code-mixed texts. We provide a comparison of 2 approaches to annotating code-mixed data a) annotation of the texts from the two languages using monolingual resources from each language and b) annotation of the text through a different resource created specifically for code-mixed data. We present a comparative study of the efforts required in each approach and the final performance of the system. Based on this, we argue that it might be a better approach to develop new technologies using code-mixed data instead of monolingual, 'clean' data, especially for those languages where we do not have significant tools and technologies available till now.

1 Introduction

Code-mixing and code-switching in multilingual societies are two of the most well-studied phenomena within the field of sociolinguistics (Gumperz, 1964; Auer, 1995; Myers-Scotton, 1997; Muysken, 2000; Cardenas-Claros and Isharyanti, 2009). Generally, code-mixing is considered intra-sentential in the sense that it refers to mixing of words, phrases or clauses within the same sentence while code-switching is inter-sentential or even inter-clausal in the sense that one switches to the other language while speaking. In this paper, we will use code-mixing to refer to both these phenomena.

While code-mixing is a very well-studied phenomena within the field of theoretical linguistics, there have been few works computational modelling of code-mixing. In the past of few years, with the explosion of social media and an urgent need to process the social media data, we have seen quite a few efforts at modelling, automatic identification and processing of code-mixing (most notable among them being (Solorio and Liu, 2008a; Solorio and Liu, 2008b; Nguyen and Dogruoz, 2013; Das and Gambck, 2014; Barman et al., 2014; Vyas et al., 2014) and several others in the two workshops on computational approaches to code-mixing).

In this paper, we discuss the development of a part-of-speech tagger for English-Assamese code-mixed data and also present a comparative study of two different approaches to annotating code-mixed data

a monolingual ensemble approach: reuse the already available tools for individual languages in an ensemble to process the code-mixed data and

b novel multilingual approach: develop new tools exclusively for code-mixed data from the scratch.

It is often argued that it is a much more resource-intensive task to develop separate tools for different kinds of natural language processing of code-mixed data. As such it is desirable to use the pre-existing tools that were developed for different languages for processing code-mixed texts. While this argument holds merit if the languages under consideration have sufficiently large number of tools and applications already available, which may be used. However, this is not the case for a large number of Indian

This work is licenced under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/

Proceedings of the First International Workshop on Language Cognition and Computational Models, pages 94–103
Santa Fe, New Mexico, United States, August 20, 2018.
https://doi.org/10.18653/v1/P17

languages, including the major ones. Barring a few exceptions, there is hardly any basic technologies available for most of the Indian languages. In such a situation, developing tools and technologies for code-mixed, multilingual texts might prove to be more efficient and effective than those for monolingual texts. Also, it might be the case that the tools developed for code-mixed texts work better with monolingual texts in comparison to the performance of the tools developed for monolingual texts used with code-mixed texts. In this paper, we discuss the challenges and issues of both the approaches to processing code-mixed data and also discuss the comparative performance of both the approaches and argue for a rather provocative stand - it will be a better and more fruitful idea to develop technologies based on a multilingual, code-mixed data instead of what is considered 'clean', monolingual data not only because code-mixed data will become norm in the near future but also because these technologies might prove to be 'overall' better performing ones of the two.

2 Corpus Collection and Annotation

Since there is no previous corpus available for Assamese-English code-mixed data, we collected a large corpus of such data from four different public Facebook pages:

- https://www.facebook.com/AAZGFC.Official
- https://www.facebook.com/Mr.Rajkumar007
- https://www.facebook.com/ZUBEENsOFFICIAL
- https://www.facebook.com/teenagersofassamm

These facebook pages contain adequate amount of Assamese-English code-mixed data. The dataset was annotated at the word-level with 2 kinds of information language and part-of-speech. These annotations were carried out with an aim to develop two kinds of system

a language identification system, which is needed for annotating the dataset with individual monolingual taggers of the languages in the text and

b part-of-speech tagger for the code-mixed texts.

The annotation schemes are discussed in the following subsections. We also discuss the collection and annotation of monolingual English and Assamese datasets for the experiments.

2.1 Language Annotation of the Dataset

The data was annotated with both the information about the language at the word-level as well as with the part-of-speech tags. The tagset used for the language annotation is given in Table 1.

The data was annotated at 3 levels Matrix Language, Fragment Language and Word-level Code-mixing (WLCM). Matrix language refers to the language of the whole comment and it may be monolingual (Assamese or English), code-mixed (Mix), universal (UNIV) and named entity (NE). If the language is neither of these three, it is annotated as Other - it allows for further annotation of these comments in the dataset with specific language. Fragment language is the word-level annotation of the language and it was annotated with the same set of languages as the matrix language, except Mix. WLCM refers to the phenomenon where the root form of a word is in one language and the affix is in another language. In such cases, the language of the word is annotated as a combination of the two languages which makes up the word. Let us take a look at the following example -

> Thik koise..Mission china **Indiar** babey aru A Wondrous Army **Worldr** babey...kiman wait korabo aru..release diok hunkale..
>
> You are right...."Mission China" is **for India** and "A Wondrous Army" is **for the world**...How long will you make us wait....(You) release immediately..

In this comment, 'Indiar' and 'Worldr' are instances of WLCM, wehere 'India' and 'World' are English words and '-r' is the Assamese marker for benefactive here.

Sl. No.	Top Level	Language	Label
1.	Matrix Languages	1.1, 1.2, 1.3, 1.4, 1.5, 1.6	–
2.	Fragment Languages	1.1, 1.2, 1.4, 1.5, 1.6 pt	–
3.	Word-level Code-mixing	1.7, 1.8, 1.9, 1.10, 1.11, 1.12	–
1.1		Assamese	AS
1.2		English	EN
1.3		Mix	MIX
1.4		Other	OT
1.5		Universal	UNIV
1.6		Named Entity	NE
1.7		Assamese-English	AS-EN
1.8		English-Assamese	EN-AS
1.9		Assamese-Other	AS-OT
1.10		Other-Assamese	OT-AS
1.11		English-Other	EN-OT
1.12		Other-English	OT-EN

Table 1: Language Identification Tagset

2.2 Part-of-Speech Annotation of the Dataset

Universal part-of-speech tags, proposed by the Universal Dependencies was used for annotating the data with part-of-speech information. The tagset is reproduced in Table 2.

In addition to the 17 universal tags included in the Universal Dependencies tagset, 2 tags suffix and prefix - were included in the tagset. It was necessitated by the kind of data that we encountered in our dataset. There were several instances where the affixes in the Assamese text (written in Roman script) were not attached to their root. Let us take a look at an example below -

It was generally observed that the classifiers and genitive markers were not attached to their root form while writing in Roman. This could be possibly because of the lack of a standardized writing convention in a non-native script like Roman and the identification of a false word boundary by the speakers, which led them to separate the root and the affix in the texts. We did not normalize such instances and in order to annotate such fragments, the 2 new tags were introduced. The reason for not normalzsing texts like these was 2-fold - a) these could actually be an indication towards the way language is processed and word boundaries recognised by the speakers and b) in case there is a variation, it may point towards sociopragmatic usage of separating out certain kinds of 'affixes' from their roots.

All the other tags carry the same meaning as in the universal dependencies tagset. Emojis in the text were marked as Symbol.

2.3 Monolingual Assamese Dataset and Annotation

In addition to the code-mixed annotated dataset that we created, we also acquired monolingual Assamese dataset, prepared as part of Indian Languages Corpora Initiative (ILCI) and made available through Technology Development for Indian Languages (TDIL), Govt. of India. The dataset contains 2 kinds of data original Assamese texts from newspapers, magazines, etc from more than 10 different domains and translated Assamese texts (source language: Hindi) from the two domains of entertainment and agriculture. The total dataset that is currently available consists of 52,000 part-of-speech annotated sentences. However, we use only a small portion of the dataset for this study. The data was annotated using the Bureau of Indian Standards (BIS) tagset that has been declared the national standard for annotating Indian languages data. However, since all other datasets used in the experiments have been annotated with Universal Dependencies tagset, it was necessary that Assamese dataset also uses the same tagset. Since there is no Assamese dataset annotated with Universal Dependencies POS tagset available, we developed a simple mapper to map the tags of BIS tagset to those of Universal Dependencies. Since BIS tagset is

Sl. No.	Category	Label
1.	Noun	NOUN
2.	Proper Noun	PROPN
3.	Pronoun	PRON
4.	Adjective	ADJ
5.	Adverb	ADV
6.	Verb	VERB
7.	Auxiliary	AUX
8.	Adposition	ADP
9.	Subordinating Conjunction	SCONJ
10.	Coordinating Conjunction	CCONJ
11.	Determiner	DET
12.	Interjection	INTJ
13.	Numeral	NUM
14.	Particle	PART
15.	Punctuation	PUNCT
16.	Symbol	SYM
17.	Other	X
18.	Suffix	SUFFIX
19.	Prefix	PREFIX

Table 2: Part-of-Speech Tagset

Takei....	etiya	Raj	da'i	break	**tu**	dilei	hol	aru...
INTJ	now	Raj	brother-NOM	break	**CLF**	give-EMP	happen	and

Now, may rajda give the break and thats it

more fine-grained than the UD tagset, it was a rather simple task to map the tags from BIS to UD tagset. The mapping is given in Table 3.

While for the most part the mapping was quite straightforward and simple to implement, there were a couple of instances where the differing guidelines made the things a little difficult. One was the case of general quantifiers. Generally, quantifiers occur at the position of demonstrative in a syntactic structure and this is probably the reason why quantifiers are classified as determiners and not numerals in UD. However, in the BIS tagset, it is grouped with the numerals. Similarly, BIS tagset do not have determiners as a separate category but they have demonstratives which do not appear in UD. The reasons again seem to be syntactic - since UD is more generally designed for syntactic parsing, the POS categories are accordingly defined. In both these cases, we followed the UD guidelines while mapping since that is the tagset which is being mapped into.

In addition to these, UD does not have echo-word at POS level - it has been included as a morphological feature, which is pretty obvious. Since it was not possible to map this to any POS category in UD, we used a new category called 'suffix' to map echo-word to. It could be argued that it is not a POS category but it is also not meant to be so. It is only a placeholder such that it could be properly handled at the morphemic level. Furthermore, since we are using this category in annotating the social media data also, it also provided some kind of consistency.

Aside from all this, what was surprising was that the Assamese dataset was not annotated with the information about 'classifiers'. Since BIS tagset provides for a category called 'classifier' and Assamese is quite rich in terms of classifiers, this category must have been included. However since it was not present in our dataset, we have not mapped it to any other category. In any case, it does appear in some dataset, like echo-word, it could also be mapped to 'suffix'.

Sl. No.	BIS Category	BIS Tag	UD Category	UD Tag
1.	Common Noun	N_NN	Noun	NOUN
2.	Nloc	N_NST	Noun	NOUN
3.	Proper Noun	N_NNP	Proper Noun	PROPN
4.	Personal Pronoun	PR_PRP	Pronoun	PRON
5.	Reflexive	PR_PRF	Pronoun	PRON
6.	Relative Pronoun	PR_PRL	Pronoun	PRON
7.	Reciprocal	PR_PRC	Pronoun	PRON
8.	Wh-word	PR_PRQ	Pronoun	PRON
9.	Indefinite Pronoun	PR_PRI	Pronoun	PRON
10.	Deictic Demonstrative	DM_DMD	Determiner	DET
11.	Relative Demonstrative	DM_DMR	Determiner	DET
12.	Wh-word Demonstrative	DM_DMQ	Determiner	DET
13.	Indefinite Demonstrative	DM_DMI	Determiner	DET
14.	Main Verb	V_VM	Verb	VERB
15.	Auxiliary	V_VAUX	Auxiliary	AUX
16.	Adjective	JJ	Adjective	ADJ
17.	Adverb	RB	Adverb	ADV
18.	Postposition	PSP	Adposition	ADP
19.	Subordinating Conjunction	CC_CCS	Subordinating Conjunction	SCONJ
20.	Coordinating Conjunction	CC_CCD	Coordinating Conjunction	CCONJ
21.	Default Particle	RP_RPD	Particle	PART
22.	Interjection	RP_INJ	Interjection	INTJ
23.	Intensifier	RP_INTF	Particle	PART
24.	Negation	RP_NEG	Particle	PART
25.	General Quantifier	QT_QTF	Determiner	DET
26.	Cardinal Quantifier	QT_QTC	Numeral	NUM
27.	Ordinal Quantifier	QT_QTO	Numeral	NUM
28.	Punctuation	RD_PUNC	Punctuation	PUNCT
29.	Symbol	RD_SYM	Symbol	SYM
30.	Foreign Word	RD_RDF	Other	X
31.	Unknown	RD_UNK	Other	X
31.	Echo-word	RD_ECH	Suffix	SUFFIX

Table 3: Mapping of BIS Assamese tagset to Universal Dependencies tagset

2.4 Monolingual English Dataset and Annotation

For English, the monolingual annotated dataset was obtained from the dataset provided for CoNLL 2018 shared task. The dataset was annotated using the Universal Dependencies tagset. We used the Universal Dependencies English Web Treebank v2.2, which consists of 16,622 sentences, taken from five genres of web media: weblogs, newsgroups, emails, reviews, and Yahoo! answers. As with the Assamese dataset, we used only a randomly sampled small subset of this dataset for our experiments.

3 Challenges and Issues: A comparison

Both the approaches to processing code-mixed multilingual documents monolingual ensemble approach as well as novel multilingual approach come with their own unique set of challenges and they need to be handled in their own way. We shall discuss some of the challenges that we faced and how we solved those.

3.1 Requirement of helper technologies

The monolingual ensemble approach assumes the availability of the helper technologies for the languages in the text. For our research, these technologies include the following

a Word-level language identification system: It is the first pre-requisite of the monolingual method that the language of the tokens be correctly identified so that they could be processed by the systems of respective languages. For our experiments, we used the system described in (Bora and Kumar, 2018).

b Part-of-Speech taggers: We developed part-of-speech taggers for English as well as Assamese using the monolingual data for the respective languages mentioned in the previous section.

c Transliteration System: Like most of the other Indian languages, a significant proportion of Assamese is written in Roman script over the web. However, the monolingual systems are developed to work on the texts in native script. As such a transliteration module is required to transliterate the roman texts into native script so that the monolingual taggers could process the data. For our experiments, since Roman Assamese transliteration system is not available, we used Roman Bangla transliteration system, which is a very close approximation because of the mostly shared script of the two languages.

The novel multilingual approach, however, only requires that a new part-of-speech tagger be trained for the complete dataset.

3.2 Different Standards and Formats

As we have been seen in the previous section, English and Assamese have used two different 'standards' for part-of-speech annotation of the dataset. In this case, since both the tagsets have been quite standardised and have been in use for a lot of languages, it was a relatively simple task to map those. However, in a lot of different tasks, there have been a large number of different tagsets and annotation schemes, with a glaring lack of a standard, to the extent that every language uses a different annotation scheme. In such a situation, mapping of tagsets such that the tagsets of all the languages in the code-mixed data are same, might become a herculean task and, in fact, may not be completely possible in certain instances.

However, developing a new system using the code-mixed dataset rules out any such requirement of mapping different tagsets for different tasks.

3.3 Error Propagation

It is a commonly known fact that the greater the number of systems involved in a pipeline, greater is the error as the error from one system propagates and multiplies through different stages in the pipeline. As we have seen, the monolingual ensemble approach requires that at least 2 (and sometimes even more) systems work in the pipeline. This is likely to increase the error count. In the following section, we will see the extent to which an ensemble system leads to huge errors in the whole pipeline.

4 Experiments and Discussion

We developed 3 different part-of-speech taggers - Assamese, English and Code-mixed - as part of our experiments. All the 3 taggers were trained on a dataset of approximately 1,700 sentences each. We divide the dataset into train:test ratio of 90:10. The train set is used for training a Linear SVM classifier using 5-fold cross-validation. We tune only C hyperparamter of the classifier and arrive at the best classifier using Grid Search technique. We use scikit-learn library (in Python) for all our experiments. The following set of features gave the best performance for all the three classifiers -

Word-level Features: We used the current word, previous 2 words and next 2 words as features.

Tag-level Features: We used the tags of previous 2 words as features.

Character-level Features: We used the first three characters (prefixes) and last three character (suffixes) as features for training

Boolean Features: In addition to the above features, we also used the following additional features has_hyphen (1 if the word has hyphen in it), is_first / is_second (1 if the word is the first / second word in the sentence), is_last / is_second_last (1 if the word is the last / second last word in the sentence) and is_numeric (if the word is a number).

We will be releasing the dataset and the models trained during the experiments for further research as well as reproducibility of our results

These classifiers were tested in 3 different ways to assess the relative performance of the systems developed using the two different approaches to processing code-mixed data. These are discussed in the following subsections.

4.1 Same train-test dataset

This is the classical testing of the classifiers where we test the classifiers on the dataset of the same language as it was trained on. Thus the classifier trained on Assamese monolingual dataset was tested on Assamese monolingual dataset and so on. The test results set a benchmark to compare the loss of performance when tested on the other datasets. The performance of the classifiers is summarised in Table 4

Train Set	Test Set	Precision	Recall	F1
Assamese	Assamese	0.90	0.90	0.90
English	English	0.88	0.88	0.88
Code-mixed	Code-Mixed	0.85	0.84	0.84

Table 4: Performance of part-of-speech taggers tested on the dataset of same language

As we could see, the classifier for code-mixed data performs the worst. This is not very surprising given the low amount of data that was used for training. However, with similar amount of data, the other 2 classifiers performed comparatively better. This could be attributed to the fact that the monolingual dataset is more consistent and noise-free than the code-mixed data and thus comparatively easier to fit than the code-mixed data. Moreover, it must be noted that in this case, it is not just that the code is mixed; rather the dataset is from social media and contains several other kinds of inconsistencies including non-standard spelling and punctuation, use of emoticons, presence of hyperlinks, etc. As such, the training data required for training a code-mixed classifier is more than that required for monolingual classifier, in order to achieve a comparable performance.

4.2 Train on code-mixed, test on monolingual

In this case, we used the part-of-speech tagger trained on code-mixed dataset to test on both the English as well as Assamese monolingual dataset. A comparative performance of the classifier on both the monolingual datasets as well as the code-mixed dataset is summarised in Table 5

Train Set	Test Set	Precision	Recall	F1
Code-mixed	Assamese	0.64	0.65	0.64
Code-mixed	English	0.67	0.65	0.65

Table 5: Performance of part-of-speech taggers trained on code-mixed dataset and tested on the monolingual dataset

As expected, there is a drop in the performance of the classifier when it is tested on a dataset different from the one it was trained on. In fact, it was not just a different dataset, it was trained on a dataset with a different language and consequently dataset with a large amount of vocabulary not present in the train set. Given the fact that, for a task like part-of-speech tagging, the classifier was not performing at its best, the drop in the performance is reasonable.

4.3 Train on monolingual, test on code-mixed

In this last case, we basically followed the ensemble approach of annotation where we use a pipeline of 4 different systems to annotate the code-mixed test set with part-of-speech information and evaluate it. Figure 1 shows the annotation pipeline.

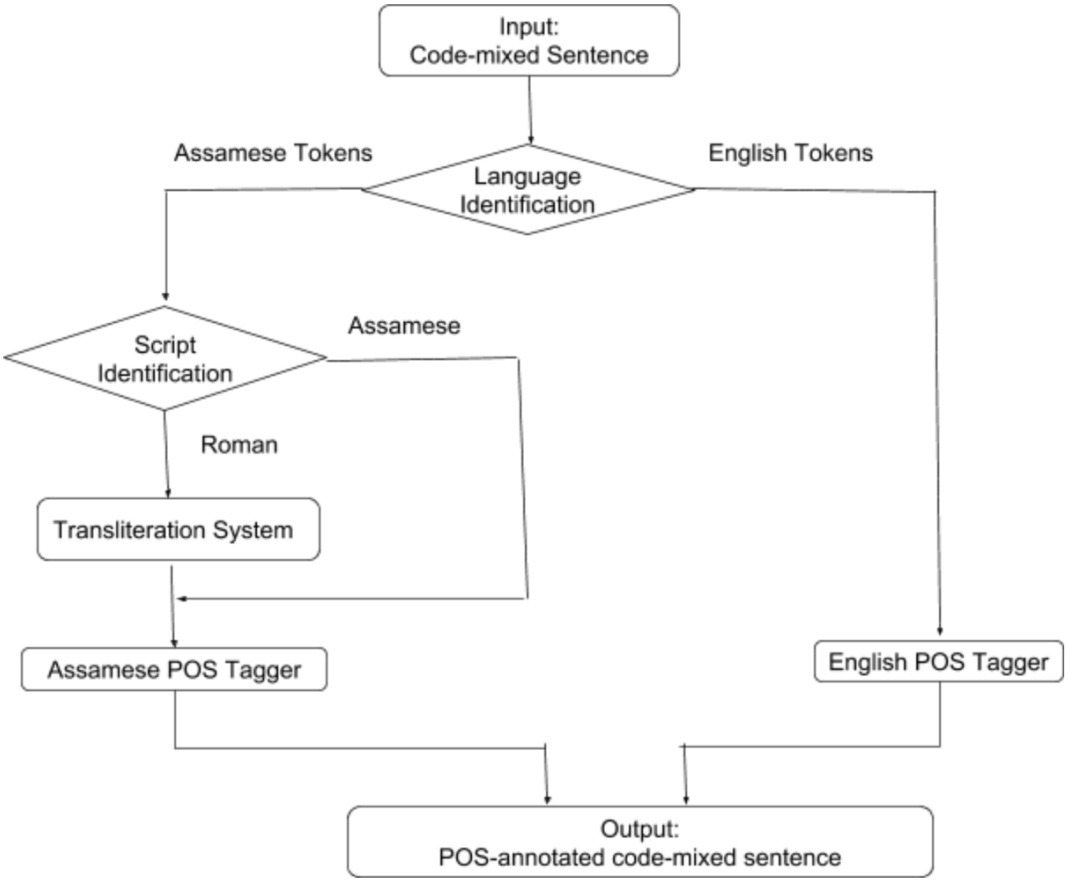

Figure 1: The annotation pipeline for code-mixed data using ensemble approach

In the first step, the test set was annotated with language tags at the word-level. Then the Assamese tokens in Roman were transliterated using Google's transliteration system for English-Bangla pair since there is no transliteration system available for Roman to Assamese. Finally in the last step, depending on whether the token is English or Assamese, the English or Assamese tagger was used to annotate it. If the token was a punctuation or an emoticon, they were marked as punctuation and symbol without using the

tagger. The performance of this system vis-a-vis the one trained on the code-mixed data is summarised in Table 6

Train Set	**Test Set**	**Precision**	**Recall**	**F1**
Code-mixed	Code-mixed	0.85	0.84	0.84
Assamese + English + OT [1]	Code-mixed	0.59	0.50	0.50

Table 6: Performance of part-of-speech taggers tested on code-mixed data

The huge drop in the performance of the classifier is pretty obvious. It is not difficult to guess the reason behind this drop. It is not just the errors made by the part-of-speech classifier but also the errors by the language identification system as well as the transliteration system (the fact that it was not even English-Assamese transliteration system and the data that we transliterated was from social media did not help either) that overall resulted in a performance like this. It would be interesting to explore how the system will perform if we assume that language identification and transliteration systems performed perfectly well. We already have the test set manually annotated with language tags and we are currently in the process of manually transliterating the test set. Once done, we will be able to report on how much the errors in each system of the pipeline add up to. However, despite this, in practical applications, we cannot expect to get manually annotated and transliterated datasets and as such in real-life we expect the system to perform as reported here.

5 Summing Up

In this paper, we have discussed the issues and challenges of using the monolingual ensemble approach over the novel multilingual approach. We argue that, given the number of technologies required for using the ensemble approach, it may not be a practical or even beneficial approach to follow if the required systems are not already available for all the languages in our dataset. On the contrary, if we are building new tools and technologies for any language, it would be highly desirable that such systems are trained on multilingual code-mixed data from the social media for some very obvious reasons. It is quite easy and quick to collect such data. Also our experiments show that training a system on code-mixed data performs relatively well on monolingual data. Moreover, while the overall annotated data required for a comparable performance on code-mixed dataset is more than that required for building a monolingual system, the overall data requirement is actually less than the overall data required for building systems for *all* the languages in the code-mixed dataset.

References

P. Auer. 1995. The pragmatics of code-switching: A sequential approach. In L. Milroy and P. Muysken, editors, *One Speaker, Two Languages: Cross-Disciplinary Perspectives on Code-Switching*, pages 115–135. Cambridge University Press, Cambridge.

U. Barman, A. Das, J. Wagner, and J. Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23.

Manas Jyoti Bora and Ritesh Kumar. 2018. Automatic word-level identification of language in assamese english hindi code-mixed data. In *4th Workshop on Indian Language Data and Resources, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 7–12.

M. Cardenas-Claros and N. Isharyanti. 2009. Code-switching and code-mixing in internet chatting: Between 'yes', 'ya', and 'si' a case study. *The JALT Call JOurnal*, 5(3):67–78.

A. Das and B. Gambck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*.

J. John Gumperz. 1964. Hindi-punjabi code-switching in delhi. In *Proceedings of the Ninth International Congress of Linguistics*, The Hague. Mouton.

P. Muysken. 2000. *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press, Cambridge.

Carol Myers-Scotton. 1997. *Duelling Languages: Grammatical Structure in Code-Switching*. Clarendon Press, Oxford.

D Nguyen and A. S. Dogruoz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 857–862.

T. Solorio and Y. Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 973–981.

T. Solorio and Y. Liu. 2008b. Parts-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060.

Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 974–979.