POS Tagging of Hindi-English Code Mixed Text from Social Media: Some Machine Learning Experiments

Royal Sequiera, Monojit Choudhury, Kalika Bali
Microsoft Research Lab India
{a-rosequ,monojitc,kalikab}@microsoft.com

Abstract

We discuss Part-of-Speech(POS) tagging of Hindi-English Code-Mixed(CM) text from social media content. We propose extensions to the existing approaches, we also present a new feature set which addresses the transliteration problem inherent in social media. We achieve an 84% accuracy with the new feature set. We show that the context and joint modeling of language detection and POS tag layers do not help in POS tagging.

1 Introduction

Code Switching (CS) and Code Mixing (CM) are natural phenomena observed in all stable multilingual societies. Code Switching refers to the co-occurrence of speech extracts belonging to two different grammatical system (Gumperz, 1982) in a single utterance. Whereas, Code Mixing denotes the usage of linguistic units of one language into an utterance that belongs to another language(Myers-Scotton, 1993). In this paper, we will use CM to refer to both of these situations.

CM is predominately a speech-level phenomenon, though with the prevalence of social media and user-generated content that are more speech-like, we now observe CM quite commonly in text as well (Crystal, 2001; Herring, 2003; Danet and Herring, 2007; Cardenas-Claros and Isharyanti, 2009). Therefore, it is imperative that we develop NLP techniques for processing of CM text to analyze the user-generated content from and cater to the needs of multilingual societies.

In the recent past, there has been some work on CM data most of which has been focused on word level language identification (Solorio and Liu, 2008a; Saha Roy et al., 2013; Gella et al., 2013) and POS tagging of CM text which is one of the first steps towards processing of CM text. Parts-of-Speech tagging is another task which has been explored to a little extent for CM text (Solorio and Liu, 2008b; Vyas et al., 2014). POS tagging of CM data is an interesting problem to study both from a practical and a theoretical perspective because it requires modeling of the grammatical structures of both the languages as well as the syntactic constraints applicable on CM.

In this paper, we explore machine learning approaches for POS tagging of Hindi (**Hi**)-English (**En**) CM text from social media. We start with replication of the experiments presented in (Vyas et al., 2014) and (Solorio and Liu, 2008b), and reconfirm their results on our dataset. Then we extend the set of features used by (Solorio and Liu, 2008b) and do several feature selection experiments. Finally, we also propose and conduct a joint language labeling and POS-tagging task. Our experiments show that while there is marginal improvement due to use of certain additional features, joint modeling significantly hurts the results.

The rest of the paper is organized as follows: Sec 2 discusses the related work; in Sec 3, we introduce some basic concepts and definitions. Dataset is described in Sec 4 and the baseline experiments in Sec 5. Sec 6 discusses experiments with additional features and Sec 7 the joint modeling approach. Finally, we summarize our work and conclude in Sec 8.

2 Related Work

Parts-of-Speech tagging for monolingual text has been studied extensively with an accuracy as high as 97.3% for some languages (Toutanova et al., 2015). However, not much work has been done on POS tagging of CM text. Solorio and Liu (2008b) were the first to introduce this problem through their work on POS tagging of Spanish-English CM text collected by recording a conversation between three bilingual speakers and then manually transcribing the recording. They presented a set of rule-based methods which included tagging the

D S Sharma, R Sangal and E Sherly. Proc. of the 12th Intl. Conference on Natural Language Processing, pages 237–246,
Trivandrum, India. December 2015. ©2015 NLP Association of India (NLPAI)

Previous Work/Approaches	*CM*	*Social Media*	*Machine Learning*	*Transliteration & Spelling Variation*
(Solorio and Liu, 2008b)	✓	✗	✓	✗
(Gimpel et al., 2011)	✗	✓	✓	✓
(Vyas et al., 2014)	✓	✓	*	✓
(Jamatia and Das, 2014)	**	✓	✓	✗

Table 1: A comparison of the previous work
* Was not a primary focus but POS tagging of CM text was discussed.
** Did not use machine learning for developing a CM POS tagger but the individual POS taggers were trained using supervised machine learning.

Hi POS Tag	Universal POS Tag
NC	NOUN
NP	NOUN
NV	NOUN
NST	NOUN
VM	VERB
VAUX	VERB
PPR	PRON
PRF	PRON
PRC	PRON
PRL	PRON
PWH	PRON
JJ	ADJ
JQ	ADJ
DAB	PRON
DRL	PRON
DWH	PRON
AMN	ADV
ALC	ADV
PP	ADP
CSB	CONJ
CCL	CONJ
CX	PRT
CCD	CONJ
PU	.
RDX	NUM
RDF	X
RDS	.

Table 2: Mapping from **Hi** POS tag set to the Universal POS tagset

text through an English and a Spanish monolingual tagger and then choosing one of the two tags for a word based on some heuristics that used (a) the confidence scores of the taggers, (b) the lemma of the words, and (c) the language of the word as detected by several language detection techniques. They extend their framework by learning to predict the tag per word based on the output of the two monolingual taggers and several other features such as the confidence scores, language labels and the word itself. Naïve Bayes, SVM, Logic Boost and J48 were explored in their experimental setup. The machine learning based techniques achieved a word level tagging accuracy of around 93.5%, which is nearly a 4% improvment over the best of the rule-based systems. Note that since the speech conversations were manually transcribed, this dataset did not contain any spelling variations or transliteration, which are typical of social media text.

More recently, Vyas et al. (2014) presented an initial study on POS tagging of Hindi-English CM social media text. Apart from code-mixing, social media text poses other challenges as well, including transliteration (i.e., Romanization of Indic language words), intentional and unintentional spelling variations, short and ungrammatical text, etc. The authors created a corpora with multi-level annotations to represent the POS tag on the first level, language label on the second level and transliteration of the Hindi tokens in the final level. A simple language detection based heuristic was employed where first the text was divided into chunks of tokens belonging to a language, and then each chunk was tagged by the POS tagger for that language. Language detection and transliteration was carried out by a system described in Gella et al. (2013). Three different sets of experiments were conducted to study the effects of language detection and transliteration on the accuracy of POS tagging. With gold standard language labels and transliteration, a word level tagging accuracy of 79.02% has been reported, which is a good 15% improvement over the case where both language detection and transliteration were done automatically. This study not only highlights the importance of accurate language detection and transliteration for POS tagging of social media text, but also establishes the inherent hardness of the problem. Clearly, POS tagging of CM text cannot be solved by juxtaposition of two monolingual POS taggers.

Another recent study by Jamatia and Das (2015) briefly mentions POS tagging of Hindi-English CM tweets, though the primary focus of their work was tagging of monolingual Hindi tweets. They report 63.5% word level tagging accuracy for some Random Forest based pilot experiments on 400 CM utterances (all romanized) from Facebook and Twitter. On the other hand, the authors report around 87% accuracy on monolingual Hindi tweets written in Devanagari. Thus, this work also illustrates the hardness of POS tagging transliterated and CM social media text.

In this context, it is useful to note that there has been quite a few studies on POS tagging of monolingual social media content for English and a few other languages. Gimpel et al. (2011) proposed one of the first POS taggers for English tweets. A tag set for representing the POS tags in social media content was presented. They used a CRF tagger with arbitrary local features in a log-linear model adaptation. The feature set included context cues such as the the presence of digits or hyphens and capitalization in a word, and features representing suffixes upto length 3. The augmented feature set also amassed external linguistic resources, the domain specific properties of data and unlabeled in-domain data. An accuracy of 89.95% was reported.

Owoputi et al. (2013) proposed an improvement over this original Twitter POS tagger. In addition to the unsupervised word clustering features, the tagger exploits lexical features, which escalates the accuracy from aforementioned 90% to 93%. However, none of these studies consider code-mixing.

	#Matrices	
Type matrix	***Vyas et al. (2014)***	***Jamatia et al.(2014)***
#HiMono	126	9
#HiCM	61	213
#EnMono	189	0
#EnCM	30	0
Total Matrices	406	222
	#Tokens	
#Tokens	4,157	5,633
% of matrices with CM	22.4	95.94

Table 3: Data set statistics

Table 1 presents a comparative summary of the aforementioned approaches.

3 Basic Concepts and Annotation

With the advent of social media, we are now witnessing considerable amount of CM in text data, which is primarily due to the fact that social media data, and more generally other forms of CM such as e-mails,blogs are speech-like comments Bali et al. (2014). The following example for instance, demonstrates CS and CM in social media content:

Dude I think u should try again caz ye [this] tera [your] fault nahi [not] hai [is]. ye [this] CBSE walo [people] ki [of] fault hai [is].

The above utterance is an instance of both CM and CS. It is code-switched because the first part of the sentence "*Dude I think u should try again caz*" is in **En** matrix whereas the rest of the sentence is in **Hi** matrix. It is also code-mixed as there are **En** words such as fault is embedded within the **Hi** matrix.

More formally,

1. ***Matrix Language:*** The language governing the grammar of an utterance is called as the matrix language of the utterance. (plural: *matrices*)

2. ***Embedded Language:*** Refers to the language of the words that are not in the matrix language, but nevertheless, are *embedded* in the utterances.

3. ***Switch Points:*** Suppose that $q \; :< w_1 w_2 w_3 \ldots w_n \; >$ is an utterance; i is a

switch point if and only if the language of the word w_i is different from w_{i+1}

Normalization is defined as the process of transforming an input text that might contain non-standard spellings and informal syntax to a standard or canonical representation of the spellings and grammar. If the language is not written in the script that is normally used, then the process of normalization would also involve back-transliteration from the non-native script to canonical word forms in the native script.

POS tagging of a CM text in social media is a challenging task due to the following reasons:

1. **Paucity of annotated CM data:** Annotating any data is a laborious task. However, there are further complications in CM data annotations. A bi-lingual speaker who is proficient at both the languages who also has matching linguistic background may be required for annotating POS tags of CM text. Although crowd-sourcing can be an alternative approach, it comes with the risk of inaccurate annotations. For this reason, crowd sourcing is not a very viable alternative for this kind of annotation (Jamatia and Das, 2014).

2. **Transliteration of tokens:** Traditionally, Indic languages are written in their native script. But, due to various socio-technical reasons, the computer mediated channels have been observing a lot of romanized content Sowmya et al. (2010). Bali et al. (2014) showed that less than 5% of the **Hi** content popular in social media are in native script. This can appear to be a challenging task for identifying the language of the words and thereby POS tagging of such words.

4 Data Set

For this study, we use data from two different sources. The first set was created by (Vyas et al., 2014). The corpora contains posts and comments belonging to Facebook pages of various celebrities and the BBC Hindi news page. The second source ocomes from Jamatia et al. (2015). The data is acquired from @BBCHindi and @aajtak using a Java based Twitter API. The statistics for each source is summarized by Table 3. There is a total of 628 utterances with 9,790 tokens and 48.4% of data features CM.

Vyas et al. (2014) mention the matrix language to be either **Hi** or **En**. But, for the sake of granular analysis we divide matrix language into four parts: Hindi-Monolingual (**HiMono**), Hindi-Code Mixed (**HiCM**), English-Monolingual (**EnMono**) and English-Code Mixed (**EnCM**). We also follow a multilevel annotation and the same illustrated by figure 1.

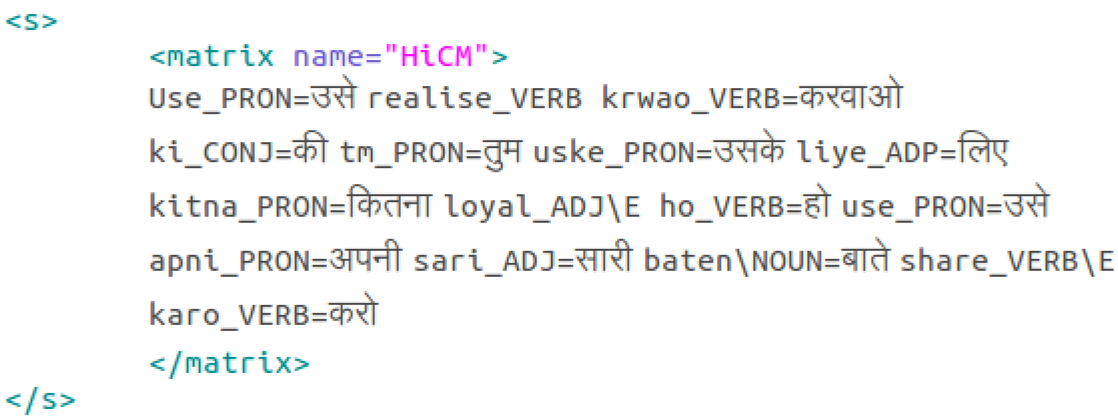
For our study, we use the tags generated by the individual POS taggers. But as the tag sets for **Hi** (Sankaran et al., 2008) (also known as **ILPOST** tagset) and **En** (Marcus et al., 1993) tagger are of different, we map both the tag sets to a universal POS tag set as proposed by(Petrov et al., 2011). The mapping from **En** tag set to the universal tag set has already been proposed by (Petrov et al., 2011); we present a mapping from **Hi** tag set to the universal POS tag set which is shown in Table 2.

5 Baseline Experiments

In this section we present baseline experiments for the VGSBC and SL model which is essentially the replication of the approaches proposes by Vyas et al. (2014) and Solorio and Liu (2008b) respectively. However, in the next section, we discuss additional features and extensions to the existing models.

5.1 VGSBC Baseline Experiments

Initially, we conduct the experiments proposed by (Vyas et al., 2014), and we refer to this model as VGSBC model following the names of the authors. The VGSBC model divides the text into chunks of tokens having same language. After which, the **Hi** chunks are tagged by the **Hi** POS tagger and **En** chunks are tagged by **En** POS tagger. The model presents three different experiments: The first experiment uses gold language labels(LL) and gold normalization (HN) of the token. On the other hand, the second experiment uses gold language labels but automated normalization of the tokens which helps one to individually study the effect of gold standard normalization. Finally, machine generated language labels and transliteration is used to establish the their combined role. We implement an n-gram based language identifier as proposed by (Gella et al., 2013; King and Abney, 2013a). For generating back-transliterations, we use a transliteration system inspired by (Gella et al., 2013). For **Hi**, a



```
<s>
        <matrix name="HiCM">
        Use_PRON=उसे realise_VERB krwao_VERB=करवाओ
        ki_CONJ=की tm_PRON=तुम uske_PRON=उसके liye_ADP=लिए
        kitna_PRON=कितना loyal_ADJ\E ho_VERB=हो use_PRON=उसे
        apni_PRON=अपनी sari_ADJ=सारी baten\NOUN=बाते share_VERB\E
        karo_VERB=करो
        </matrix>
</s>
```

Figure 1: An annotation example

	Matrix : Hindi			*Matrix : English*			*Overall*
	HiMono	*HiCM*	*Overall*	*EnMono*	*EnCM*	*Overall*	
Gold Std LL, Gold Std HN	0.794	0.764	0.769	0.827	0.817	0.825	0.782
Gold Std LL, Machine HN	0.775	0.759	0.762	0.754	0.722	0.749	0.759
Machine LL, Machine HN	0.759	0.741	0.744	0.544	0.556	0.546	0.698

Table 4: VGSBC results on the data set

CRF++ based Hindi POS Tagger is used which can be downloaded from http://nltr.org/snltr-software/ and for **En**, we use the Twitter POS tagger (Owoputi et al., 2013) which is also freely available at http://www.ark.cs.cmu.edu/TweetNLP/. The tagger has an inbuilt tokenizer and normalizer specifically fabricated to handle social media content.

Result: Table 4 shows the results on our test data set using the VGSBC model. The accuracy is the highest (78.2%) when gold language labels and gold normalization is used. Also, VGSBC with gold language labels and gold normalization performs 6% better than VGSBC with machine generated language labels and automated normalization.

5.2 SL Baseline Experiments

The VGSBC model proposes a modest approach to POS tagging. As each chunk is tagged separately by either of the monolingual taggers, crucial information that can be captured by the other tagger is missed. Moreover, as the entire utterance is not tagged by the tagger, a right POS tag cannot be determined for every chunk. In other words, only completely monolingual utterances will be tagged appropriately by such an approach. Although the model uses language labels as well as the normalization of the tokens, it does not employ any machine learning algorithms to train a CM POS tagger. In contrast, SL model presents an approach which leverages the tags spawned by both the taggers. Furthermore, an entire utterance is passed to the individual taggers and either of the tags (from **Hi** POS tagger or **En** POS tagger) is chosen based on various heuristics:

5.2.1 Using Individual Taggers

We run the CM text through the individual taggers and measure the accuracy with respect to each tagger. As mentioned earlier, we pass the entire utterance to both taggers and then compute the accuracy for each tagger so as to determine how well monolingual the taggers work on CM text.

5.2.2 Using Language Labels

We also use the language labels to select the appropriate POS of the word in a CM text:

1. **Automatic Language Detection:** The language labels of words in the CM text was de-

	$Matrix : Hindi$			$Matrix : English$			$Overall$
	$HiMono$	$HiCM$	$Overall$	$EnMono$	$EnCM$	$Overall$	
ILPOST	0.800	0.722	0.735	0.192	0.252	0.207	0.610
Twitter	0.286	0.353	0.342	0.827	0.789	0.821	0.455
Automatic Language Detection	0.777	0.817	0.811	0.772	0.718	0.765	0.799
Human Language Detection	0.800	0.823	0.820	0.827	0.789	0.821	0.821
Oracle							0.864

Table 5: SL baseline accuracy

	$Matrix : Hindi$			$Matrix : English$			$Overall$
	$HiMono$	$HiCM$	$Overall$	$EnMono$	$EnCM$	$Overall$	
Naive Bayes	0.756	0.748	0.753	0.798	0.794	0.797	0.777
MaxEnt	0.795	0.795	0.795	0.862	0.849	0.861	0.831

Table 6: SL machine learning experiment

termined by using an n-gram based language identifier suggested by(Gella et al., 2013). The POS of the word is chosen based on its language label i.e. if the word is identified to be **En**, the output by **En** POS tagger will be considered as the POS of the word and similarly the POS of the **Hi** POS tagger is taken as the POS of the word if the language of the word is identified to be **Hi**.

2. **Gold Standard Language Detection:** The word level language labels of the CM text was done by a human annotator and the gold language labels of the words in CM text were used to choose the right POS of the word.

5.2.3 Oracle

Finally, to establish a baseline accuracy, we check if one of the POS tags generated by the individual POS taggers matches the gold POS tag and accordingly calculate the accuracy. In other words, Oracle gives the accuracy when the right tag is chosen from the available POS tags of monolingual POS taggers.

5.2.4 SL Baseline Results

As shown in Table 5, the monolingual taggers fail miserably as the CM text contains words that are foreign to the monolingual taggers. Therefore, an accuracy of 45.5% and 61.1% is obtained for **En** and **Hi** respectively. Unsurprisingly, the accuracy is high when the matrices are monolingual and when the appropriate POS tagger is used. When the gold language labels are used, the accuracy of the POS tags increases dramatically and an accuracy of 82.1% is obtained.

5.3 SL Machine Learning Experiments

We also conduct the machine learning experiments proposed by Solorio and Liu with the following features:**En** POS tag, **Hi** POS tag, POS confidence and the current token.

Due to the lack of information such as the confidence score and the tagger lemma, we could not conduct the experiments verbatim. However, the above mentioned features are closest to the features proposed by Solorio and Liu that we could replicate.

We use MALLET (MAchine Learning LanguagE Toolkit) which can be downloaded from `http://mallet.cs.umass.edu/` for training the CM POS tagger. As the data set is relatively smaller, a 10-fold cross validation was used for all the experiments. The experiment is conducted in three steps. We trained a Naïve Bayes and a MaxEnt based model aforementioned in the SL model.

5.3.1 Results

The accuracy obtained for each algorithm is as shown in Table 6. MaxEnt algorithm outperforms

$Context$	3	2	1	0
$Accuracy$	0.798	0.812	0.827	0.837

Table 7: Effect of context on learning

Scheme	***Accuracy***
Without Normalization (SL)	0.831
With Automated Normalization	0.834
With Gold Normalization	0.840

Table 8: Effect of Normalization on Accuracy

the Naive Bayes algorithm by 7%. Therefore, MaxEnt is chosen for further experiments.

5.4 Comparison of the Models

Let us now compare the numbers in table 4 and 5. When the gold standard language labels are used, the VGSBC model performs with 75.9% accuracy whereas the SL model has an accuracy of 82.1%. Similarly, when automated language labels are used, the VGSBC model works with an accuracy of 69.8% but the SL model performs with 82.1% accuracy. This clearly shows that tagging an utterance first and later choosing the POS tag based on the language label works better than dividing the utterance into chunks based on the language labels and then passing the chunks to the POS taggers.

The VGSBC and SL model show improvement in the accuracy (6% and 2% respectively) when gold language labels are used. The improvement in the accuracy can be owed to the fact that automatic language detection of a CM data plays a very important role and is far from a solved problem (Solorio et al., 2014). Further, the Oracle accuracy on our data set was found to be 86.4%.

The SL machine learning model works better than the SL baseline model with automatic language detection by 3% but is less than Oracle almost by the same margin. It is also seen that the machine learning model better than the SL baseline model (by 1%) with human language identification.

6 Additional Features and Joint Modeling

Although the SL models performs better than the VGSBC model, it does not use several features that the modern taggers use today. For instance, the SL tagger does not utilize the contextual features, normalization features, sub-word features such as whether the first letter in a word is capitalized and so on. In this section, we propose additional features to the existing feature set used by the SL model and also propose extensions to the existing models.

6.1 The Feature Set

In addition to the feature set proposed by the SL model, we use the feature set proposed by Chittaranjan et al. (2014). The final augmented feature set is shown below:

Monolingual POS tagger Features: The output generated by the individual POS taggers mapped to the universal POS tag set is used as features. The confidence of the taggers is also used as a feature but as ILPOST does not generate a confidence score per tag, the confidence score generated by the Twitter tagger alone is used.

Normalization Feature: The words are normalized to their native script. That is, if the word is an **En** word, its standard form is used. Similarly, if the word is an **Hi** word, its gold transliteration is used. The rationale behind using this feature is that the text is written only in Roman script and is prone to non-standard spellings.

Contextual Features: These features include context cues such as the current token, an array of previous and next words, previous token and previous tag.

Capitalization Features: Indicates if the token is capitalized or not. These features signal an instance of Named Entity which could be a proper noun.

Special Character Features: Social media posts generally contain special characters like @, # etc.

Lexicon Features: They capture the existence of a token in lexicons. The lexicons include a **Hi** dictionary of most frequent words, **En** dictionary of most frequent words, a list of common NEs and a list of common acronyms.

Language Identifier Scores: We used character n-grams (King and Abney, 2013b) to train two language identifiers. We trained two separate language identifiers to identify words of **Hi** and **En**. We trained each such classifier with 5000 cases of positive instances and 5000 cases of negative instances. The classifiers were trained on $Maximum\ Entropy (MaxEnt)$ and the proba-

Features	*Accuracy*
Context	0.837
Hi and **En** LL	0.837
Lexicons	0.837
Hi Lexicon	0.838
NE and Acronym Lexicon	0.836
Hi, NE and Acronym Lexicon	**0.840**
Punctuation	0.838
Capitalization	0.836
Hi Normalization	0.830
Current Token	0.802
En Confidence	0.836
Hi, **En** POS and **En** Confidence	0.721
Capitalization, Punctuation and Lexicon	0.836

Table 9: The ablation experiment

bility scores for each label was used as a feature for CM POS tagger.

6.2 Experiments

In this section, we discuss various experiments with the new found features. We use *MaxEnt* algorithm for all the experiments discussed in this section with a 10-fold cross validation.

6.2.1 Context Based Experiments

Firstly, we choose the Monolingual POS tagger features (POS_{Hi}, POS_{En}, $\text{Confidence}_{\text{En}}$) along with Normalization Feature (NRM) and experiment for the best context. We run the experiment on different window size of words to determine the right context that produces the best accuracy. We begin with a context of three words (previous three words and next three words), i.e. window size = 3 and reduce the size by 1 for each experiment.

6.2.2 Normalization Experiments

We also propose extensions to SL model by using normalization of tokens as one of the features. Initially, we train a CM POS tagger with the aforementioned features without any normalization. This is exactly in line with what SL model proposes. In the next set of experiments, we use a machine transliteration of the token as the feature. Finally, we use the gold transliteration as the feature to compare the gain with accuracy when the transliteration of the token is perfectly known.

6.3 Results

In the context based experiment, we observe that the accuracy increases consistently as the context decreases which is surprising as the context should have helped the learner. We also find that the word alone (no context) is the optimal context for the tagger. This is summarized by Table 7.

We see that the normalization slightly betters the performance. There is 1% increase in the accuracy when gold normalization is used. Table 8 exemplifies the gain in accuracy with the addition of normalization layer.

After choosing the right context size, we conduct a set of ablation experiments to study the effect of a set of features on the accuracy. We selectively turn off some of the features and observe the corresponding change in the accuracy for such a set of features. We conducted an elaborate feature selection step which is recapitulated by Table 9. From the table we observe that when the POS tags of the monolingual taggers are turned off, the accuracy drops steeply indicating that POS tags of the monolingual taggers contribute the most to the learning. The highest accuracy obtained is 84% when **Hi**, NE and Acronym lexicons are ablated and we call the corresponding model as SL++ model.

7 Joint Modeling

As the above proposed approaches use two separate layers viz. a language identification layer and a normalization layer, we propose a joint modeling of both the layer and investigate the resulting tagger. Our reasoning is that the the errors propagating through the layers can be avoided by designing a joint modeling system. To implement the above proposal, we come up with a new tag system which is a product of the POS tag set and language label set.

Let $\rho : POS_1, POS_2, ... \ POS_{12}$ be the POS tags.

Let $\Lambda : L_1, L_2, L_3$ be the language labels.

Then, the new tag set T obtained is defined as:

$\mathcal{T} : \rho \times \Lambda$

We choose the best model from Table 9 and run the joint modeling experiments. This approach yields an accuracy of 77.33%.

	Matrix : Hindi			*Matrix : English*			*Overall*
	HiMono	*HiCM*	*Overall*	*EnMono*	*EnCM*	*Overall*	
VGSBC	0.803	0.813	0.815	0.827	0.817	0.817	0.812
SL	0.849	0.861	0.851	0.926	0.930	0.911	0.913
SL++	0.852	0.861	0.859	0.930	0.908	0.915	0.926

Table 10: Matrix level accuracy of the systems

8 Discussion and Conclusion

In this study, we have seen that the VGSBC model performs poorly in comparison to the SL model baselines. This shows that passing the entire utterance to the monolingual POS tagger and then choosing the appropriate tags based on the language label works better than passing the monolingual fragments of the utterance to the respective monolingual tagger. We also see that the machine learning based technique described (Solorio and Liu, 2008) performs much better than all the baselines that only use some heuristics on top of the monolingual taggers. This essentially reestablishes the findings by Solorio and Liu (2008b).

Our extended feature experiments show that the context features do not help. In fact, the accuracy consistently increases as the context is narrowed down all the way until no context is used. Further, the SL model with the augmented features provides only marginal improvements. We believe that this is due to the paucity of training and test data. To verify this proposition, we trained and tested the VGSBC, SL and SL++ models on our entire dataset, the results of which are shown in 10. It is seen that the accuracy obtained for each model on the training data is consistently higher and that too by a large margin than when we did k-fold validation (all our previous experiments). Thus, it is clear that with context and other features, the models are over-fitting to the data and as a result we see no benefit. We do believe that some of the features, especially the context is useful and experiments on larger training set, will be able to bring this fact out. Similarly, the joint modeling approach also shows a degraded performance probably because of larger number of tags and insufficient data to learn from.

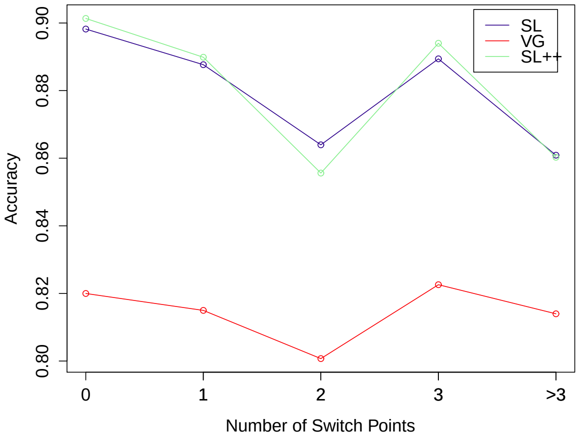


Figure 2: An annotation example

In order to understand the pain points of CM text processing, we also analyzed the correlation between the number of switch points in a sentence and the accuracy of POS tagging. Figure 2 shows the plot of number of switch points (so 0 essentially refers to monolingual utterances) versus the word level accuracy of the tagger averaged on all test sentences with that many switch points. We see that with the increase of number of switch points, the accuracy falls dramatically for up to 2 switch points. However, the accuracy for three switch points is higher than one or two switch points. When we investigated into this, we found that there are very few examples in our test set with 3 or more switch points and as a result it is impossible to make any conclusions from there.

In future, we would like to address the data scarcity problem through a multi-pronged approach of (a) annotating more data, (b) using unsupervised machine learning techniques, and (c) better learning from monolingual utterances. Another promising direction of research could be to model this problem as a structured output prediction rather than a pointwise classification problem.

Acknowledgment

We would like to thank Amitava Das,IIIT-Sri City and Anupam Jamatia, NIT Agartala for sharing their annotated dataset. We are also grateful to Rafiya Begum, MSR India for her help with reviewing the annotations.

References

Kalika Bali, Yogarshi Vyas, Jatin Sharma, and Monojit Choudhury. 2014. ''i am borrowing ya mixing?" an analysis of English-Hindi code mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP*.

Mónica Stella Cardenas-Claros and Neny Isharyanti. 2009. Code-switching and code-mixing in internet chatting: Between yes, ya, and si a case study. In *The JALT CALL Journal, 5*.

Gokul Chittaranjan, Yogrshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identication using crf : Code-switching shared task report of msr india system. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*.

David Crystal. 2001. *Language and the Internet*. Cambridge University Press.

Brenda Danet and Susan Herring. 2007. *The Multilingual Internet: Language, Culture, and Communication Online*. Oxford University Press., New York.

Spandana Gella, Jatin Sharma, and Kalika Bali. 2013. Query word labeling and back transliteration for indian languages: Shared task system description. In *FIRE Working Notes*.

Kevin Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of ACL*.

John J. Gumperz. 1982. *Discourse Strategies*. Oxford University Press.

Susan Herring, editor. 2003. *Media and Language Change*. Special issue of Journal of Historical Pragmatics 4:1.

Anupam Jamatia and Amitava Das. 2014. Part-of-speech tagging system for Hindi social media text on twitter. In *Proceedings of the First Workshop on Language Technologies for Indian Social Media, ICON*.

Anupam Jamatia, Bjrn Gambck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *In the Proceeding of 10th Recent Advances of Natural Language Processing (RANLP)*.

B King and S. Abney. 2013a. Labelling the languages of the world in mixed-language documents using weakly supervised methods. In *Proceedings of NACL-HLT, 2013*.

Ben King and Steven Abney. 2013b. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of NAACL-HLT*, pages 1110–1119.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Carol Myers-Scotton. 1993. *Dueling Languages: Grammatical Structure in Code-Switching*. Claredon, Oxford.

Olutobi Owoputi, Brendan OConnor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview and datasets of fire 2013 track on transliterated search. In *FIRE Working Notes*.

Bhaskaran Sankaran, Kalika Bali, Monojit Choudhury, Tanmoy Bhattacharya, Pushpak Bhattacharyya, Girish Nath Jha, S. Rajendran, K. Saravanan, L. Sobha, and K. V. Subbarao. 2008. A common parts-of-speech tagset framework for indian languages. In *Proceedings of LREC*.

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Empirical Methods in natural Language Processing*.

Thamar Solorio and Yang Liu. 2008b. Parts-of-speech tagging for English-Spanish code-switched text. In *Proceedings of the Empirical Methods in natural Language Processing*.

Thamar Solorio, Elizabeth Blair, Suraj Maharajan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alson Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*.

V. B. Sowmya, Monojit Choudhury, Kalika Bali, Tirthankar Dasgupta, and Anupam Basu. 2010. Resource creation for training and testing of transliteration systems for indian languages. In *Proceedings of the Language Resource and Evaluation Conference (LREC)*.

Kristina Toutanova, Dan Kleina, Christopher Manning, and Yoram Singer. 2015. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In the Proceeding of 10th Recent Advances of Natural Language Processing (RANLP)*.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of English-Hindi code-mixed social media content. In *Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP*.