Language Identification and Analysis of Code-Switched Social Media Text

Deepthi Mave, Suraj Maharjan and Thamar Solorio
Department of Computer Science
University of Houston
{dmave, smaharjan2}@uh.edu, solorio@cs.uh.edu

Abstract

In this paper, we detail our work on comparing different word-level language identification systems for code-switched Hindi-English data and a standard Spanish-English dataset. In this regard, we build a new code-switched dataset for Hindi-English. To understand the code-switching patterns in these language pairs, we investigate different code-switching metrics. We find that the CRF model outperforms the neural network based models by a margin of 2-5 percentage points for Spanish-English and 3-5 percentage points for Hindi-English.

1 Introduction

Code-switching occurs when a person switches between two or more languages in a single instance of spoken or written communication (Gumperz, 1982; Myers-Scotton, 1997). Code-switching instances are prevalent in modern informal communications between multilingual individuals specially, in social media platforms such as Facebook and Twitter. Given this prevalence of code-switching, there is value in automatic processing and understanding of such data. Language identification at the word level is the first step in computational modeling of code-switched data. Language identification is important for a wide variety of end user applications such as information extraction systems, voice assistant interfaces, machine translation, as well as for tools to assist language assessment in bilingual children (Gupta et al., 2014; Chandu et al., 2017; Roy et al., 2013). Language detection, in addition, enables sociolinguistics and pragmatic studies of code-switching behavior.

Code-switching in speech is well studied in linguistics, psycholinguistic and sociolinguistics (Sankoff, 1970; Lipski, 1978; Poplack, 1980; Gumperz, 1982; Auer, 1984; Myers-Scotton, 1997, 2002). The alternation of languages across sentence boundaries is known as code-switching and the alternation within a sentence is known as code-mixing. In this paper we will refer to both instances as code-switching and differentiate between the types of code switching when necessary. Table 1 shows examples of code-switching for Hindi-English and Spanish-English.

Example 1
Good morning sir*ji*, *aaj ka* weather *kaisa hai*? (Good morning sir, How is the weather today?)
Example 2
Styling day *trabajando con* @username *vestuario para* #ElFactorX *y soy hoy chofer*. I will get you there in pieces im a Safe Driver. (Styling day working with @username wardrobe for #ElFactorX and today I am a driver. I will get you there in pieces im a Safe Driver.)

Table 1: Example 1 shows code-switching between Hindi-English and Example 2 between Spanish-English (Molina et al., 2016).

Word level language identification of code-switched text is inherently difficult. First, a single code-switched instance can have mixing at the sentence or clause level, the word level, and even at the sub-word level (e.g. sir-*ji*, *chapathi*-s). Second, the typology of the languages involved in switching and their inter-relatedness further increase the task complexity. For example, a shared Latin influence on Spanish and English results in lexical relatedness (Smith, 2001; August et al., 2002), making Spanish-English language identification harder than Hindi-English. Third, in spite of the fact that Hindi has a native script (Devanagari), most of the Hindi social media text is

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 51–61
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

transliterated. Transliteration is conversion of a text from one script to another. In the case of Hindi, text is converted from native, Devanagari to Roman script. Due to lack of standardization in transliteration, a single Hindi word can have multiple surface forms (e.g. *Humara, Hamara, Hamaaraa* etc.). Some Hindi words can take the same surface form as an English word. The words *'hi'* (an auxiliary verb), *'is'* (this), and *'us'* (that) are some examples. Finally, the characteristics of social media text such as non-standard spelling, contractions, and not strictly adhering to the grammar of the language adds to the list of challenges.

In this work, we make three contributions. First, we build a new code-switched dataset for Hindi-English (HIN-ENG) language pair from Facebook public pages and Twitter. Second, we investigate different code-switching metrics for Hindi-English and a standard Spanish-English (SPA-ENG) dataset. Third, we compare a traditional machine learning model - conditional random field (CRF), and two recurrent neural network (RNN) based systems, for word-level language identification of the above language pairs. In contrast to the CRF model, the RNN-based systems do not involve language specific resources or sophisticated feature engineering. We test these models, first for each of the language pairs individually, and then for a corpus with both the language pairs combined.

Among the language identification systems, the CRF model outperforms both the RNN-based systems across language pairs. When both the language pairs are combined, the result from the best performing model (CRF) is 25% points higher than the baseline system. The RNN-based models also give reasonable results.

2 Related Work

Over the last decade several researchers have explored word-level language identification for different language pairs and dialect varieties. The FIRE shared task series - (Roy et al., 2013; Choudhury et al., 2014; Sequiera et al., 2015b) focuses on language identification of code-mixed search queries in English and Indian languages for information retrieval. We use a larger set of labels compared to these tasks. The First and Second Shared Task on Language Identification in Code-Switched Data (Solorio et al., 2014; Molina et al., 2016) show the necessity for automatic processing of code-switched text and report comparison of different language identification systems. The best system from the second iteration of these shared tasks uses a logistic regression model and reports a token-level F1-score of 97.3% for SPA-ENG. Our results are competitive with this score. Das and Gambäck (2014) use a dictionary based method and SVM model with various features for Hindi-English and Bengali-English. Their system achieves an F1-score of 79% for Hindi-English. Barman et al. (2014) create a new dataset and study code mixing between the three languages - English, Hindi, and Bengali using CRF and SVM models. In another work, Gella et al. (2014) build a language detection system for synthetically created code-mixed dataset for 28 languages. Similar to some of the works in the above mentioned papers, we model the language detection task as a sequence labeling problem and explore combinations of several features using the CRF model, but we use a larger set of labels. We obtain significantly higher performance for the Hindi-English language pair than Das and Gambäck (2014).

Along with the traditional machine learning approach, some researchers have also used models based on artificial neural networks. Chang and Lin (2014) use an RNN architecture with pre-trained word2vec embeddings for SPA-ENG and the Nepali-English datasets from the First Shared Task on Language Identification in Code-Switched Data. Samih et al. (2016) build an LSTM based neural network architecture for SPA-ENG and MSA-DA datasets from the Second Shared Task on Language Identification in Code-Switched Data. Their model combines word and character representations initialized with pre-trained word2vec embeddings. We replicate their model with *softmax* output layer for SPA-ENG and run similar experiments for HIN-ENG, as well as with both the corpora combined. Our result for SPA-ENG match that of Samih et al. (2016).

3 Data

We use the SPA-ENG dataset from the EMNLP Code-Switching Workshop 2016. This data is collected from Twitter, based on the geographical areas with strong presence of Spanish and English bilingual speakers - California, Texas, Miami, and New York (Solorio et al., 2014; Molina et al., 2016). The labels used are summarized in Table 2. The hashtags are treated as a word and are la-

Label	Description	HIN-ENG (%)	SPA-ENG (%)
lang1	English words only	57.764	38.258
lang2	Hindi/Spanish words only	20.418	40.579
ne	Proper names	6.582	1.935
other	Symbols, usernames, emoticons	14.807	18.952
mixed	Words partially in both the languages	0.04	0.018
ambiguous	Can't determine whether English or Hindi/Spanish	0.009	0.137
fw	Words is not English neither Hindi/Spanish	0.369	0.01
unk	Unrecognizable word	0.012	0.11

Table 2: A brief description of the labels and label distribution for HIN-ENG and SPA-ENG datasets.

beled accordingly.

Corpus Creation for Hindi-English. For the HIN-ENG corpus, we consider Facebook pages of prominent public figures from India. Hindi-English bilingual users are highly active in these pages (Bali et al., 2014). We crawl posts and their comments from the Facebook public pages of various sports-persons, political figures, and movie stars. We also crawl random tweets from geographical locations Mumbai and Delhi using the Twitter API. From the crawled posts, we remove the posts in native scripts, and remove duplicate and promotional posts. We filter the posts containing URLs and those with less than 3 words.

Language Pair	Tweets	Tokens	Unique
Pair	(Posts)		Tokens (%)
SPA-ENG	25,130	294,261	35,153 (11.95)
HIN-ENG	7,421	146,722	23,998 (16.36)

Table 3: Corpus statistics for the language pairs. Token ratio is the percentage of the total tokens that are unique. A higher token ratio implies a richer corpus vocabulary.

We follow EMNLP 2016 shared task annotation guidelines and use a semi-automatic approach to annotate the data. The labels are reviewed and corrected with the help of in-lab annotators. The inter-annotator agreement score over approximately $4,000$ tokens is 0.935. A portion of the Facebook dataset is annotated using the English lexicon and Hindi transliterated pairs.[1,2] We use pattern matching rules to label punctuations, emoticons, and usernames. These labels are then corrected manually for *ne*, *fw*, *mixed*, *ambiguous*, and *unk* labels. We also make use of two existing datasets - Facebook dataset from ICON2016 POS tagging shared task and the dataset from (Sequiera et al., 2015a).[3] We manually map the labels of these data sets to labels in Table 2. We train a character n-gram based CRF model using the above mentioned three datasets (see Section 5.2) and predict the labels for all the posts crawled from Facebook and the random tweets from Twitter. From these, we identify the posts predicted as code-switched, correct the labels where necessary, and add them to the final dataset. The F1-weighted score for this model is close to 96 percent.

4 Code-Switching Analysis

In this section we provide some descriptive statistics about the corpora to understand the language distribution and language-relatedness. Table 4 shows the language distribution at post (tweet) level. The SPA-ENG dataset has a balanced distribution where as, in the HIN-ENG dataset majority of the instances are in English. The below statistics show that both the datasets have a good amount of code-switched instances to train and test the language identification systems. Table 2 presents the label-wise token distribution for the datasets. For HIN-ENG, majority of the words (58%) are in English, 20% are in Hindi, and 7% are named-entities. The SPA-ENG dataset in comparison has a balanced distribution of the two languages with 38% of the words in English, 41% in

Language	CS	*lang1*	*lang2*	*other*
Pairs	Instances			
HIN-ENG	43.62	51.77	4.02	0.60
SPA-ENG	34.75	33.53	28.94	2.77

Table 4: Post-level language distribution in the datasets. Column 5 corresponds to the instances that do not have any words with language tags. *lang1*: ENG, *lang2*: HIN/SPA.

[1] http://wortschatz.uni-leipzig.de/en/download

[2] http://cse.iitkgp.ac.in/resgrp/cnerg/qa/fire13translit/

[3] http://amitavadas.com/Code-Mixing.html

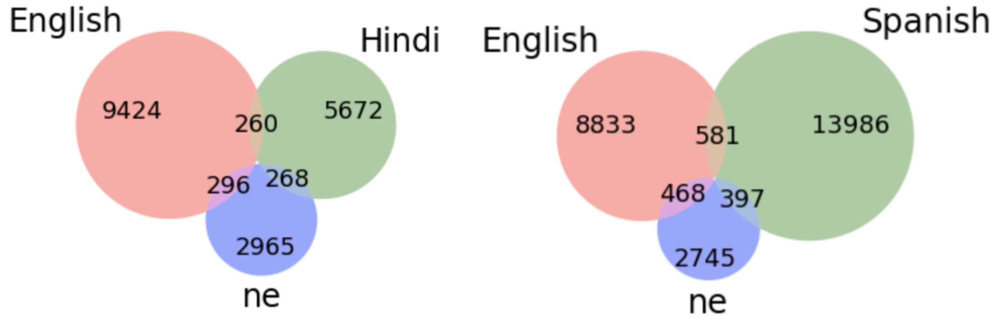


Figure 1: Vocabulary overlap for labels *lang1, lang2*, and *ne* for HIN-ENG and SPAN-ENG.

Spanish, and 2% are named-entities. The higher instances of the named-entities in the HIN-ENG dataset is a result of the way the data is sourced.

Figure 1 shows the overlap between the tokens belonging to *lang1*, *lang2*, and *ne*. These overlaps introduce ambiguity for the automatic labeling task. Around 2.5% of the Hindi words in HIN-ENG share the same spelling as some English words because of transliteration of Hindi text to Roman script. In comparison, there is a 6% overlap between Spanish and English words in the SPA-ENG dataset (e.g. *no, a, final*). This indicates higher degree of lexical relatedness between Spanish and English as compared to Hindi and English. The overlap between language words and named-entities is due to words such as *university* and *united*. These words can be part of names of organizations, movie titles or song titles and can also be used as language constructs in either of the languages.

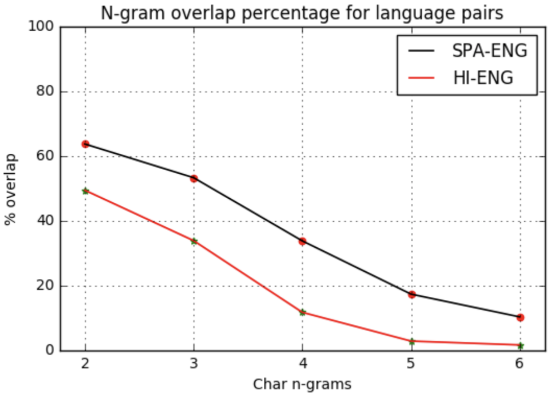


Figure 2: Plot of character n-grams overlap between the languages in the datasets, for n = 2, 3, 4, 5 and 6.

In another analysis, we explore the similarity in character n-gram profiles of the languages involved (Maharjan et al., 2015). A higher similarity in the character n-grams increases the difficulty of the task. We generate character n-grams of length 2 to 6 from the language vocabularies of each corpora. We show the plot of the character n-gram overlaps for HIN-ENG and SPA-ENG in Figure 2. As expected, the overlap decreases rapidly with increase in n-gram length. The SPA-ENG n-gram overlap is higher than that of HIN-ENG for all n-gram lengths. This trend is consistent with the results in Figure 1. To further understand the complexity involved, for an n-gram occurring in both the languages, we calculate the probability of that n-gram being a part of an English word in the corpus. A probability closer to 50% indicates higher ambiguity in classifying that n-gram. We find that a significant fraction (25%) of these shared n-grams, averaged over all n-gram lengths, appear in the range 40%-60%.

5 Code-Switching Metrics

The code-switching behavior can be different depending on the medium of communication, context of language use, topic, authors (or speakers), and the languages being mixed among other factors. We compute 3 different metrics to understand code-switching patterns in our datasets, as well as to rationalize the performance of the language identification models.

M-Index: Multilingual index is a word-count-based measure that quantifies the inequality of the language tags distribution in a corpus of at least two languages (Barnett et al., 2000). Equation (1) defines the *M-Index* as:

$$M - Index = \frac{1 - \sum p_j^2}{(k-1)\sum p_j^2} \quad (1)$$

where k is the total number of languages and p_j is the total number of words in the language j over the total number of words in the corpus. The value ranges between 0 and 1 where, a value of 0 corresponds to a monolingual corpus and 1 corresponds to a corpus with equal number of tokens from each language.

Integration Index: Integration Index is the approximate probability that any given token in the corpus is a switch point (Guzman et al., 2016; Guzmán et al., 2017). Given a corpus composed of tokens tagged by language $\{l_j\}$ where i ranges from 1 to $n-1$, the size of the corpus. The *I-index* is computed as follows:

$$I - Index = \frac{1}{n-1} \sum_{1 \leq i=j-1 \leq n-1} S(l_i, l_j) \quad (2)$$

where $S(l_i, l_j)$ = 1 if $l_i \neq l_j$ and 0 otherwise. For a corpus with n tokens, there are $n-1$ possible switch points. It quantifies the frequency of code-switching in a corpus.

Code-Mixing Index: At the utterance level, this is computed by finding the most frequent language in the utterance and then counting the frequency of the words belonging to all other languages present (Gambäck and Das, 2014). It is calculated using:

$$CMI = \frac{\sum_{i=1}^{n}(w_i) - \max(w_i)}{n-u} \quad (3)$$

where $\sum_{i=1}^{n}(w_i)$ is the sum over number of words for all N languages in the utterance, $\max(w_i)$ is the highest number of words present from any language, n is the total number of tokens, and u is the number of language independent tokens. Here, we consider the labels *lang1, lang2*, and *fw* as language words and the rest as *other*. The range of CMI value is $[0, 100)$. If an utterance has language independent tokens or only monolingual tokens, then the corresponding CMI value is 0. A higher value of CMI indicates higher level of mixing between the languages. *CMI-all* is an average over all utterances in the corpus and *CMI-mixed* is an average over only code-switched instances.

Language Pairs	**M-Index**	**CMI-all**	**CMI-Mixed**	**I-Index**
HIN-ENG	0.582	8.564	22.229	0.070
SPA-ENG	0.998	7.685	22.114	0.058

Table 5: CS Metrics for the datasets.

SPA-ENG has higher *M-Index* (Table 5) value indicating a balanced ratio of words from the two languages. This is consistent with the distribution of language words in the datasets (Table 2). The differences in *CMI-all* between HIN-ENG and SPA-ENG is about 0.9 percentage points and 0.1 percentage points for *CMI-mixed*. The higher difference for *CMI-all* could be because of the higher percentage of code-switched instances (9%) in HIN-ENG as compared to SPA-ENG (Table 4). Considering *CMI-mixed* and *I-Index* metrics together, it is evident that HIN-ENG has more language mixing and higher number of code-switching points than SPA-ENG. This is because HIN-ENG has more instances that have multiple word insertions. In SPA-ENG, instances with word insertion at more than one place in an utterance are less frequent. We also observe that a larger majority of code-switching happens between language words in HIN-ENG (76%) than in SPA-ENG (69%). For example, a number of Hindi word insertions are due to the use of the honorary article *ji* with an address form (Sir/Madam). In general, observing more code-switching in HIN-ENG is due to the fact that code-switching between Hindi and English is very widespread in India (Parshad et al., 2016; Bali et al., 2014).

6 Language Identification Models

We provide below a brief description of each of the models used.

CRF: Language identification is a sequence labeling task where the label of a token in a sequence is correlated with the labels of its neighboring tokens. So we use CRF - a sequence labeling model to capture the structure in the data. We explore different language independent features such as character n-grams, word unigram, morphological features, affixes, and contextual information for the language pairs. For each word, we generate character n-grams of length 1 to 5 and filter them based on a minimum threshold frequency of 5. To capture the morphological information of the tokens, we use binary features - is digit, is special character, is all capital, is title case, begins with @ character, has accent character (for SPA-ENG only) and has apostrophe.

We also use language dependent resources like lexicons and monolingual parts-of-speech (POS) taggers. For HIN-ENG, we use three different lexicons - Leipzig corpus for English, FIRE 2013 transliterated Hindi word pairs, and lexically normalized dictionary from Han et al. (2012) and the output of Twitter POS tagger and CRF++

based Hindi POS tagger.[4,5] For SPA-ENG, we use Leipzig corpus Spanish along with the other two lexicons mentioned above and the output from monolingual TreeTaggers for Spanish and English.[6]

Bidirectional LSTM: Long Short Term Memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are a variation of recurrent neural networks (RNNs), that address the vanishing gradient issue (Hochreiter, 1998) by extending RNNs with memory cells. A shortcoming of LSTM is that only the previous history in a sequence can be utilized. In a sequence labeling task like language identification, it is helpful to use the future context given in the sequence. Bidirectional LSTM (BLSTM) networks can access both the preceding and succeeding contexts by involving two separate hidden layers. These networks can capture the long distance relations in the sequence efficiently, in both directions. We build an end-to-end sequence model with a single BLSTM layer layer (Figure 3).

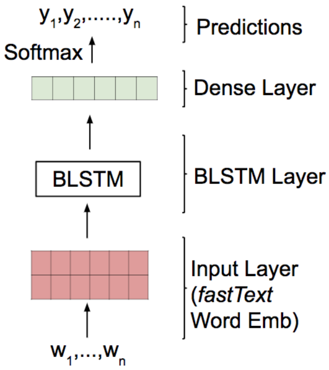


Figure 3: Bidirectional LSTM Model.

Word-Character LSTM: This model is a replication of the model proposed by Samih et al. (2016) (Figure 4). The input layer in this model has word and character embeddings. The latter are used to capture morphological features of a word. We use two LSTMs to learn fixed-dimensional representations from the embedding layers. At the output layer, we apply a *softmax* over the concatenated word and character vectors to obtain the token label. Unlike the BLSTM model, here current token and the neighboring tokens are considered to predict the label for the current token. We replace the emoticons in the dataset with a placeholder character to reduce the vocabulary size and as a result reduce the dimension of character embeddings. This decreases the number of trainable model parameters and thereby mitigates overfitting to some extent.

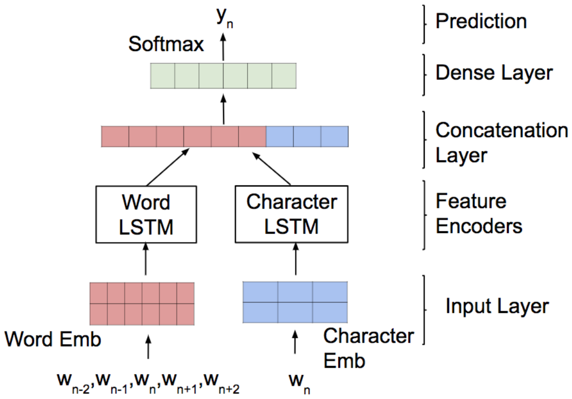


Figure 4: Word-Character LSTM Model. The input layer of word-char LSTM is initialized with *fastText* word embeddings.

7 Experiments and Results

For CRF, we run experiments with different combinations of hand-crafted features discussed in the previous section. We run three different sets of experiments- with no contextual information, and with surrounding words of context window sizes 1 and 2. Table 6 and Table 7 shows results from these experiments.

For the RNN-based systems, we use pre-trained *fastText* word embeddings.[7] We learn the embeddings using a large monolingual corpus for each of the languages and a smaller code-switched corpus for the language pairs. The rationale for using a large monolingual data is that it is readily available and that it can account for the different contexts in which words appear in different languages - thus providing an accurate separation between the languages. We train three separate sets of embeddings each for SPA-ENG, HIN-ENG, and SPA-ENG + HIN-ENG. The embeddings for SPA-ENG are trained by combining a portion of English Gigaword corpus (Graff et al., 2003) and Spanish Gigaword corpus (Graff, 2006), and a subset of tweets from Samih et al. (2016). For HIN-ENG, we combine a portion of English Gigaword corpus, transliterated Hindi monolingual corpus, and Facebook posts that contain code-switching. All

[4] http://www.cs.cmu.edu/ ark/TweetNLP/

[5] http://nltr.org/snltr-software/

[6] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

[7] https://fasttext.cc/

Experiments	**Context-0**	**Context-1**	**Context-2**
Baseline	85.02	-	-
Word + 1 to 5 char n-grams (1)	96.89	96.79	96.77
(1) + word form (2)	96.95	96.77	96.78
(2) + affixes (3)	96.96	96.84	96.84
(3) + lexicons (4)	97.07	97.03	97.12
(4) + POS tags	97.05	**97.16**	97.11
(4) + Univ POS tags	97.1	97.15	97.12

Table 6: Token-level F1-weighted score of the CRF model for different feature combinations for HIN-ENG.

Experiments	**Context-0**	**Context-1**	**Context-2**
Baseline	83.17	-	-
Word + 1 to 5 char n-grams (1)	97.02	96.81	96.82
(1) + word form (2)	97.21	97.09	97.01
(2) + affixes (3)	97.17	97.07	97.06
(3) + lexicons (4)	97.31	97.19	97.16
(4) + POS tags	97.24	97.19	97.17
(4) + Univ POS tags	**97.25**	97.19	97.21

Table 7: Token-level F1-weighted score of the CRF model for different feature combinations for SPA-ENG.

these corpora are used to train the embeddings for SPA-ENG + HIN-ENG. This helps to capture the word usage in the context of each language and eliminates the ambiguity for the words that have same surface form in multiple languages. We train 300-dimension embedding vectors using *fastText* skip-gram model for 250 epochs with a learning rate of 0.001 and a minimum word count threshold of 5.

For BLSTM model, we initialize the embedding layer with the pre-trained *fastText* word embeddings and feed the output sequence from this layer to the BLSTM layer. At the output layer a *softmax* activation function is applied over the hidden representation learned in the BLSTM layer. For word-char model, we initialize the word embedding matrix with *fastText* embeddings and use random initialization for character embedding matrix. We train both the RNN-based models by optimizing the cross entropy objective function with *Adam* (Kingma and Ba, 2014) optimizer. We use dropout masks after BLSTM layer in BLSTM model, LSTM layers in word-char model, and embedding layer in each model to mitigate overfitting. The reported BLSTM model and word-char models have hidden units of size 80 and 100 respectively in the LSTM layers. For word-char model, for each token we try a neighboring token window size of 1, 2, and 3. The context window size of 2 gives better results and is reported here.

System	**SPA-ENG**	**HIN-ENG**	**SPA-ENG + HIN-ENG**
Baseline	83.17	85.02	71.49
CRF (Context-2)	**97.06**	**96.84**	**96.37**
BLSTM	92.22	93.9	88.7
Word-char LSTM	95.46	92.19	90.1

Table 8: Token-level F1-weighted score for language identification systems.

Multiple Language Pair Experiment. We use the models described in Section 6 in an experiment to identify the labels for a dataset with multiple language pairs. This dataset has both Spanish-English and Hindi-English language pairs (SPA-ENG + HIN-ENG). To account for the third language, we use an additional label - *lang3* (HIN). Except for the pre-trained word embeddings, the models do not involve any language dependent feature engineering, and are easy to scale for multiple language pairs. As the word embeddings are

HIN-ENG		SPA-ENG	
Transitions	**Weights**	**Transitions**	**Weights**
unk → *unk*	9.511	*fw* → *fw*	4.731
fw → *fw*	5.800	*ne* → *ne*	2.798
ambiguous → *ambiguous*	4.630	*lang2* → *lang2*	1.464
lang2 → *lang2*	2.872	*lang2* → *ne*	1.005
ne → *ne*	2.824	*lang1* → *ne*	0.915
other → *other*	1.905	*lang2* → *mixed*	0.833
lang1 → *lang1*	1.535	*lang1* → *lang1*	0.707
other → *lang1*	0.801	*lang2* → *ambiguous*	0.625
lang1 → *other*	0.573	*other* → *other*	0.483
lang1 → *mixed*	0.353	*other* → *mixed*	0.427

Table 9: The top 10 most likely transitions learned by the best CRF model for HIN-ENG and SPA-ENG datasets.

trained mostly on monolingual data, this dependency does not constrain the systems.

7.1 Results and Evaluation

We use a simple lexicon-based model as baseline for our language identification systems. We use F1-weighted scores for model evaluations to account for the imbalance in label distributions (Table 2). All the models improve the performance over the respective baseline models by 7 to 25 percentage points. For CRF, which is the best performing model across language pairs, the current word and its character n-grams are the most important features. Adding POS tags does not improve these results by much. This could be because the POS taggers are optimized for monolingual data and their output for the code-switched data contains noise. Using contextual information improves the results for HIN-ENG, but not for SPA-ENG. In Table 8 we compare the RNN-based models and the CRF model. We consider the performance of the CRF model using only the language independent features with a context size of 2 for a fair comparison. Among the RNN-based systems, while the results are competitive overall, there is no single system that performs the best across language pairs. The BLSTM system performs better for HIN-ENG, while word-char system performs better for SPA-ENG. The BLSTM model captures long distance dependencies in a sequence and this is in line with the observation made above with the CRF model- more context helps for HIN-ENG. It is also consistent with the code-switching patterns discussed in Section 5. A majority of code-switched tweets in SPA-ENG have a single instance of word insertion and these are being miss-labeled by the models. The overall better results for SPA-ENG are because of a larger training data used.[8] The baseline results for SPA-ENG + HIN-ENG is relatively low as compared to the individual language pairs. This shows that simultaneously identifying language for multiple language pair is harder. We obtain reasonable results for these initial experiments with all the models.

Language Pair	System	lang1	lang2	ne
HIN-ENG	BLSTM	0.96	0.94	0.77
	Word-char LSTM	0.95	0.85	0.76
	CRF (Context-2)	**0.98**	**0.96**	**0.85**
SPA-ENG	BLSTM	0.89	0.95	0.32
	Word-char LSTM	0.89	0.97	0.40
	CRF (Context-2)	**0.94**	**0.98**	**0.57**

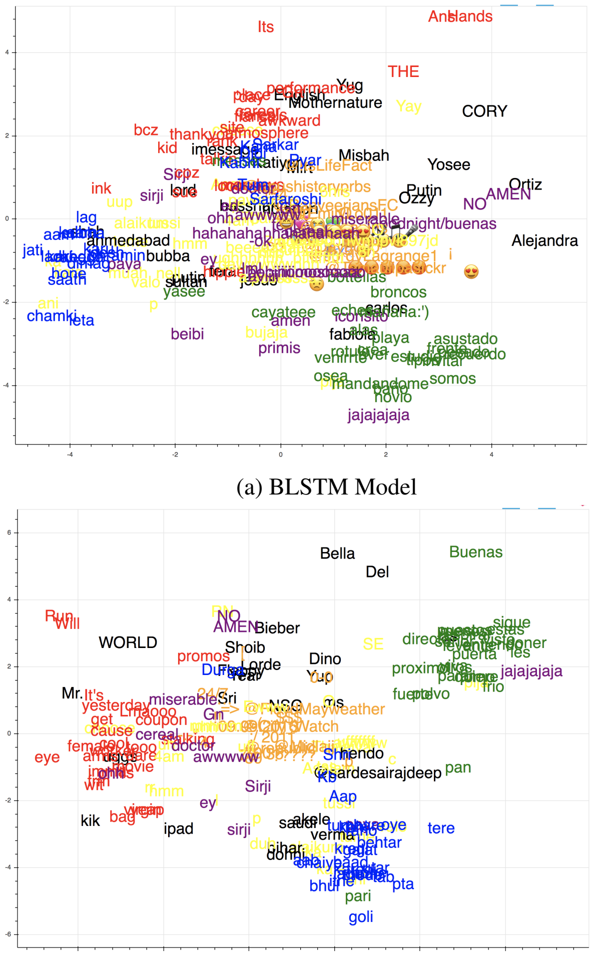
Table 10: Token-level F1-score of majority labels - *lang1, lang2* and *ne* for the models.

To understand these results better, we look at the label-wise F1-score for *lang1, lang2* and *ne* (Table 10). The F1-scores for CRF is better across the labels and the difference is significantly high for *ne*. The F1-score *ne* is relatively high for HIN-ENG, which can be attributed to the fact that around 58% of the named-entities in the test set appear in the training set. This overlap is only 17% for SPA-ENG. So, infrequent named-entities seems to be hardest to accurately label. In addition, the RNN-based models are more sensitive to amount of training samples.

Further, we examine the transitions learned by

[8] The F1-score drops by 10 percentage points for the reported experiments with the training dataset that is half in size, while maintaining the post-level language distribution.

our best CRF model for each of the language pairs (Table 9). For both language pairs, the transitions between the same languages are more likely than switching. But we also observe that the transitions from *lang1* to *lang2* and vice-versa rank higher for HIN-ENG than SPA-ENG. This is because there are fewer code-switching points in SPA-ENG as compared to HIN-ENG in these datasets.



(a) BLSTM Model

(b) Word-char LSTM Model

Figure 5: Projection of word representations learned by the neural networks model for HIN-ENG + SPA-ENG. We reduce the word vector dimensions using PCA. The mapping of labels to colors: *lang1* - red, *lang2* - green, *lang3* - blue, *ne* - black, *other* - orange, *ambiguous* - purple, *mixed* - purple, *fw* - yellow, *unk* - yellow.

We also visualize the feature representations learned by the RNN-based models by projecting the word embeddings for a randomly selected subset of words from the development datasets for SPA-ENG + HIN-ENG (Figure 5). The word-char model gives a clearer separation between the three languages, the words belonging to the labels *other* and *ne*. While the BLSTM model also provides clear separation between the language words, there is an overlap with the tokens from *other*. These results show that these models can be scaled to detect code-switching in multiple language pairs without any additional feature engineering.

8 Conclusions

The complexity of language identification of code-switched data depends on the data source, code-switching behavior, and the typology and relation between the languages involved. We find that the code-switching metrics complement each other in explaining the code-switching patterns across language pairs. The analysis of code-switching metrics shows that in our datasets Hindi-English speakers tend to switch languages more often than Spanish-English speakers. In future, it would be interesting to explore and compare the code-switching behavior of data from different sources such as movie scripts, song lyrics, and chat conversations across different language pairs.

We successfully use two different deep learning architectures without involving sophisticated feature engineering for the task and obtain competitive results. However a traditional CRF model performs better than the deep learning models for the language pairs considered. This is probably due to the amount of training data we have. The results show that word embeddings are able to capture the language separation well. Scaling these systems to identify languages in datasets with many language pairs and datasets with switching between more than two languages is a potential future direction to explore.

Acknowledgments

We would like to thank the National Science Foundation for partially supporting this work under award number 1462143 as well as the U.S. Department of Defense for the support received under grant W911NF-16-1-0422.

References

Peter Auer. 1984. *Bilingual conversation*. John Benjamins Publishing.

Diane August, Margarita Calderón, and María Carlo. 2002. Transfer of skills from Spanish to English: A study of young learners. *Washington, DC: Center for Applied Linguistics*, 24:148–158.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 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*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer 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, Doha, Qatar. Association for Computational Linguistics.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, and Mark Sebba. 2000. The LIDES Coding Manual: A document for preparing and analyzing language interaction data Version 1.1July, 1999. *International Journal of Bilingualism*, 4(2):131–271.

Khyathi Raghavi Chandu, Sai Krishna Rallabandi, Sunayana Sitaram, and Alan W Black. 2017. Speech Synthesis for Mixed-Language Navigation Instructions. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 57–61.

Joseph Chee Chang and Chu-Cheng Lin. 2014. Recurrent-neural-network for language detection on Twitter code-switching corpus. *arXiv preprint arXiv:1412.4314*.

Monojit Choudhury, Gokul Chittaranjan, Parth Gupta, and Amitava Das. 2014. Overview of FIRE 2014 track on transliterated search. *Proceedings of FIRE*, pages 68–89.

Amitava Das and Björn Gambäck. 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*, pages 378–387, Goa, India. NLP Association of India.

Björn Gambäck and Amitava Das. 2014. On measuring the complexity of code-mixing. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, pages 1–7.

Spandana Gella, Kalika Bali, and Monojit Choudhury. 2014. ye word kis lang ka hai bhai? testing the limits of word level language identification. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 368–377.

Dave Graff. 2006. LDC2006T12: Spanish Gigaword. *Linguistic Data Consortium, Philadelphia*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. LDC2003T05:English Gigaword.

John J Gumperz. 1982. *Discourse strategies*, volume 1. Cambridge University Press.

Parth Gupta, Kalika Bali, Rafael E. Banchs, Monojit Choudhury, and Paolo Rosso. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 677–686, New York, NY, USA. ACM.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. *Proc. Interspeech 2017*, pages 67–71.

Gualberto A Guzman, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2016. Simple tools for exploring variation in code-switching for linguists. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 12–20.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432, Jeju Island, Korea. Association for Computational Linguistics.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):107–116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lipski. 1978. Code-switching and the problem of bilingual competence. *Aspects of bilingualism*, pages 250–264.

Suraj Maharjan, Elizabeth Blair, Steven Bethard, and Thamar Solorio. 2015. Developing language-tagged corpora for code-switching tweets. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 72–84, Denver, Colorado, USA. Association for Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Carol Myers-Scotton. 2002. *Contact linguistics: Bilingual encounters and grammatical outcomes*. Oxford University Press on Demand.

Rana D Parshad, Suman Bhowmick, Vineeta Chand, Nitu Kumari, and Neha Sinha. 2016. What is India speaking? Exploring the Hinglish invasion. *Physica A: Statistical Mechanics and its Applications*, 449:375–389.

Shana Poplack. 1980. Sometimes ill start a sentence in Spanish y termino en espaol: toward a typology of code-switching. *Linguistics*, 18(7-8):581–618.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the FIRE 2013 track on transliterated search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, page 4. ACM.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59. Association for Computational Linguistics.

Gillian Sankoff. 1970. Social aspects of multilingualism in New Guinea. *Ph.D. thesis, McGill University*.

Royal Sequiera, Monojit Choudhury, and Kalika Bali. 2015a. POS Tagging of Hindi-English Code Mixed Text from Social Media: Some Machine Learning Experiments. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 237–246.

Royal Sequiera, Monojit Choudhury, Parth Gupta, Paolo Rosso, Shubham Kumar, Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Das Amitava Chittaranjan, Gokul, and Kunal Chakma. 2015b. Overview of fire-2015 shared task on mixed script information retrieval. In *FIRE Workshops*, volume 1587, pages 19–25.

Bernard Smith. 2001. *Learner English: A teacher's guide to interference and other problems*. Ernst Klett Sprachen.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison 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*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.