Estimating Code-Switching on Twitter with a Novel Generalized Word-Level Language Detection Technique

Shruti Rijhwani*
Language Technologies Institute
Carnegie Mellon University
srijhwan@cs.cmu.edu

Royal Sequiera*
University of Waterloo
Waterloo, Canada
rdsequie@uwaterloo.ca

Monojit Choudhury **Kalika Bali** **Chandra Sekhar Maddila**
Microsoft Research
Bangalore, India
{monojitc,kalikab,chmaddil}@microsoft.com

Abstract

Word-level language detection is necessary for analyzing code-switched text, where multiple languages could be mixed within a sentence. Existing models are restricted to code-switching between two specific languages and fail in real-world scenarios as text input rarely has a priori information on the languages used. We present a novel unsupervised word-level language detection technique for code-switched text for an arbitrarily large number of languages, which does not require any manually annotated training data. Our experiments with tweets in seven languages show a 74% relative error reduction in word-level labeling with respect to competitive baselines. We then use this system to conduct a large-scale quantitative analysis of code-switching patterns on Twitter, both global as well as region-specific, with 58M tweets.

1 Introduction

In stable multilingual societies, communication often features fluid alteration between two or more languages – a phenomenon known as *code-switching*[1] (Gumperz, 1982; Myers-Scotton, 1993). It has been studied extensively in linguistics, primarily as a speech phenomenon (Poplack, 1980; Gumperz, 1982; Myers-Scotton, 1993; Milroy and Muysken, 1995; Auer, 2013). However, the growing popularity of computer mediated communication, particularly social media, has resulted in language data in the text form which exhibits code-switching, among other speech-like characteristics (Crystal, 2001; Herring, 2003; Danet and Herring, 2007; Cardenas-Claros and Isharyanti, 2009). With the large amount of online content generated by multilingual users around the globe, it becomes necessary to design techniques to analyze mixed language, which can help not only in developing end-user applications, but also in conducting fundamental sociolinguistic studies.

Language detection (LD) is a prerequisite to several NLP techniques. Most state-of-the-art LD systems detect a single language for an entire document or sentence. Such methods often fail to detect code-switching, which can occur within a sentence. In recent times, there has been some effort to build word-level LD for code-switching between a specific pair of languages (Nguyen and Dogruöz, 2013; Elfardy et al., 2013; Solorio et al., 2014; Barman et al., 2014). However, usually user-generated text (e.g., on social media) has no prior information of the languages being used. Further, as several previous social-media based studies on multilingualism have pointed out (Kim et al., 2014; Manley, 2012), lack of general word-level LD has been a bottleneck in studying code-switching patterns in multilingual societies.

This paper proposes a novel technique for word-level LD that generalizes to an arbitrarily large set of languages. The method does not require a priori information on the specific languages (potentially more than two) being mixed in an input text as long as the languages are from a fixed (arbitrarily large) set. Training is done without any manually annotated data, while achieving accuracies comparable to language-restricted systems trained

* This work was done when the authors were affiliated with Microsoft Research.

[1] This paper uses the terms 'code-switching' and 'code-mixing' interchangeably.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1971–1982
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1180

with large amounts of labeled data. With a word-level LD accuracy of 96.3% on seven languages, this technique enabled us to analyze patterns of code-switching on Twitter, which is the second key contribution of this paper. To the best of our knowledge, this is the first quantitative study of its kind, particularly at such a large-scale.

2 Related Work

In this section, we will briefly survey the language detection techniques (see Hughes et al. (2006) and Garg et al. (2014) for comprehensive surveys), and sociolinguistic studies on multilingualism (see Nguyen et al. (2016) for a detailed survey) that were enabled by these techniques.

Early work on LD (Cavnar and Trenkle, 1994; Dunning, 1994) focused on detecting a single language for an entire document. These obtained high accuracies on well-formed text (e.g., news articles), which led to LD being considered solved (McNamee, 2005). However, there has been renewed interest with the amount of user-generated content on the web. Such text poses unique challenges such as short length, misspelling, idiomatic expressions and acronyms (Carter et al., 2013; Goldszmidt et al., 2013). Xia et al. (2009), Tromp and Pechenizkiy (2011) and Lui and Baldwin (2012) created LD systems for monolingual sentences, web pages and tweets. Zhang et al. (2016) built an unsupervised model to detect the majority language in a document. There has also been document-level LD that assigns multiple language to each document (Prager, 1999; Lui et al., 2014). However, documents were synthetically generated, restricted to inter-sentential language mixing. Also, these models do not fragment the document based on language, making language-specific analysis impossible.

Document-level or sentence-level LD does not identify code-switching accurately, which can occur within a sentence. Word-level LD systems attempt to remedy this problem. Most work has been restricted to cases where two languages, known a priori, is to be detected in the input i.e, *binary LD at the word-level*. There has been work on Dutch-Turkish (Nguyen and Dogruöz, 2013), English-Bengali (Das and Gambäck, 2014) and Standard and dialectal Arabic (Elfardy et al., 2013). King and Abney (2013) address word-level LD for bilingual documents in 30 language pairs, where the language pair is known a priori. The features for word-level LD proposed by Al-Badrashiny and Diab (2016) are language-independent, however, at any given time, the model is only trained to tag a specific language pair. There have also been two shared task series on word-level LD: FIRE (Roy et al., 2013; Choudhury et al., 2014; Sequiera et al., 2015) focused on Indian languages and the EMNLP Code-Switching Workshop (Solorio et al., 2014; Molina et al., 2016). These pairwise LD methods vary from dictionary-based to completely supervised and semi-supervised. None tackle the imminent lack of annotated data required for scaling to more than one language pair.

There has been little research on word-level LD that is not restricted to two languages. Hammarström (2007) proposed a model for multilingual LD for short texts like queries. Gella et al. (2014) designed an algorithm for word-level LD across 28 languages. Jurgens et al. (2017) use an encoder-decoder architecture for word-level LD that supports dialectal variation and code-switching. However, these studies experiment with synthetically created multilingual data, constrained either by the number of language switches permitted or to phrase-level code-switching, and are not equipped to handle the challenges posed by real-world code-switching.

Using tweet-level LD systems like the CompactLanguageDetector[2], there have been studies on multilingualism in specific cities like London (Manley, 2012) and Manchester (Bailey et al., 2013). These studies, as well as Bergsma et al. (2012), observe that existing LD systems fail on code-switched text. Kim et al. (2014) studied the linguistic behavior of bilingual Twitter users from Qatar, Switzerland and Québec, and also acknowledge that code-switching could not be studied due to the absence of appropriate LD tools.

Using word-level LD for English-Hindi (Gella et al., 2013), Bali et al. 2014 observed that as much as 17% of Indian Facebook posts had code-switching, and Rudra et al. (2016) showed that the native language is strongly preferred for expressing negative sentiment by English-Hindi bilinguals on Twitter. However, without accurate multilingual word-level LD, there have been no large-scale studies on the extent and distribution of code-switching across various communities.

[2] https://www.npmjs.com/package/cld

3 Generalized Word-level LD

We present *Generalized Word-Level Language Detection*, or GWLD, where:

- The number of supported languages can be arbitrarily large
- Any number of the supported languages can be mixed within a single input
- The languages in the input do not need to be known a priori
- Any number of language switches are allowed in the input.
- No manual annotation is required for training

Formalizing our model, let $\mathbf{w} = w_{i=1...n}$ be a natural language text consisting of a sequence of words, w_1 to w_n. For our current work, we define words to be whitespace-separated tokens (details in Sec 5). Let $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ be a set of k natural languages. We assume that each w_i can be assigned to a unique language $l_j \in \mathcal{L}$.

We also define *universal tokens* like numbers, emoticons, URLs, emails and punctuation, which do not belong to any specific natural language. Certain strings of alphabetic characters representing generic interjections or sounds, such as `oh`, `awww`, `zzz` also fall in this category. For labeling these tokens, we use an auxiliary set of labels, $\mathcal{X}_\mathcal{L} = \{xl_1, xl_2, \ldots, xl_k\}$. Labeling each universal token with a specific language l_i (using xl_i) instead of generically labeling all such tokens xl allows preserving linguistic context when a memoryless model like Hidden Markov Models (HMM) are used for tagging. Further, various NLP tasks on might require the input text, including these universal tokens, to be split by language.

For input $\mathbf{w}$, let the output from the LD system be $\mathbf{y} = y_{i=1...n}$, a sequence of labels, where $y_i \in \mathcal{L} \cup \mathcal{X}_\mathcal{L}$. $y_i = l_j$ if and only if, in the context of $\mathbf{w}$, w_i is a word from l_j. If w_i is a *universal token*, $y_i = xl_j$, when $y_{i-1} = l_j$ or $y_{i-1} = xl_j$. If w_1 is a *universal token*, $y_1 = xl_j$, where l_j is the label of the first token $\in \mathcal{L}$ in the input.

Fig. 1 shows a few examples of labeled code-switched tweets. Named entities (NE) are assigned labels according to the convention used by King and Abney (2013).

4 Method

Word-level LD is essentially a sequence labeling task. We use a Hidden Markov Model (HMM), though any other sequence labeling technique, e.g., CRFs, can be used as well.

The intuition behind the model architecture is simple – a person who is familiar with k languages can easily recognize (and also understand) the words when any of those languages are code-switched, even if s/he has never seen any mixed language text before. Analogously, *is it possible that monolingual language models, when combined, can identify code-switched text accurately?*

Imagine we have k HMMs, where the ith HMM has two states l_i and xl_i. Each state can label a word. The HMMs are independent, but they are tied to a common start state s and end state e, forming a word-level LD model for monolingual text in one of the k languages. Now, we make transitions from $l_i \rightarrow l_j$ possible, where $i \neq j$. This HMM, shown in Fig. 2, is capable of generating and consequently, labeling code-switched text between any of the k languages. The solid and dotted lines show monolingual transitions and the added code-switching transitions respectively. Fig. 2 depicts three languages, however, the number of languages can be arbitrarily large.

Obtaining word-level annotated monolingual and code-switched data is expensive and nearly infeasible for a large number of languages. Instead, we automatically create weakly-labeled monolingual text (set $\mathcal{W}$) and use it to initialize the HMM parameters. We then use Baum-Welch reestimation on unlabeled data (set $\mathcal{U}$) that has monolingual and code-switched text in their natural distribution. Sec. 5 discusses creation of $\mathcal{W}$ and $\mathcal{U}$.

4.1 Structure, Initialization and Learning

The structure of the HMM shown in Fig. 2 can be formally described using:

- Set of states, $\mathcal{S} = s \cup \mathcal{L} \cup \mathcal{X}_\mathcal{L} \cup e$
- Set of observations, $\mathcal{O}$
- Emission matrix ($|\mathcal{S}| \times |\mathcal{O}|$)
- Transition matrix ($|\mathcal{S}| \times |\mathcal{S}|$)

$\mathcal{O}$ consists of all seen events in the data, and a special symbol unk for all unseen events. We define an event as a token n-gram and we experimented with $n = 1$ to 3. It is important to mention that the n-grams do not spread over language states. We also use special start and end symbols, which are observed at states s and e respectively. Elements of $\mathcal{O}$ are effectively what the states of the HMM 'emit' or generate during decoding.

Ex(1): no\$l_2$ me\$l_2$ lebante\$l_2$ ahorita\$l_2$ cuz\$l_1$ I\$l_1$ felt\$l_1$ como\$l_2$ si\$l_2$ me\$l_2$ kemara\$l_2$ por\$l_2$ dentro\$l_2$!\$xl_2$:o\$xl_2$ Then\$l_1$ I\$l_1$ started\$l_1$ getting\$l_1$ all\$l_1$ red\$l_1$,\$xl_1$ I\$l_1$ think\$l_1$ im\$l_1$ allergic\$l_1$ a\$l_2$ algo\$l_2$

Ex(2): @XXXXX\$xl_3$ @XXXXX\$xl_3$:)\$xl_3$:)\$xl_3$:)\$xl_3$:)\$xl_3$ hahahahah\$xl_3$ alles\$l_3$ is\$l_3$ 3D\$xl_3$ voor\$l_3$ mama\$l_4$ hatta\$l_4$ 4D\$xl_4$:P\$xl_4$:P\$xl_4$:P\$xl_4$:P\$xl_4$ Havva\$l_4$ &\$xl_4$ Yusuf\$l_4$ olunca\$l_4$ misafir\$l_4$ fln\$l_4$ dinlemez\$l_4$!!\$xl_4$

Figure 1: Examples of code-switched tweets and the corresponding language labels. l_1 = English, l_2 = Spanish, l_3 = Dutch, l_4 = Turkish. Usernames have been anonymized.

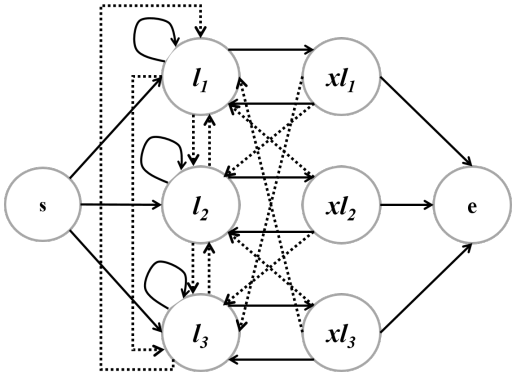

Figure 2: GWLD Hidden Markov Model. $s \rightarrow xl_i$ and $l_i \rightarrow e$ transitions omitted for clarity.

For any input, the HMM always starts in the state s. The parameters to be learned are the transition and emission matrices.

We initialize these matrices using $\mathcal{W}$. The trigram, bigram and unigram word counts from the data for each language in $\mathcal{W}$ are used to create language models (LM) with modified Kneser-Ney smoothing (Chen and Goodman, 1999). The emission values for state l_i are initialized with the respective LM probabilities for all seen n-grams. We also assign a small probability to unk. The emissions for the xl_i state are initialized using the counts of *universal tokens* for the language l_i in $\mathcal{W}$. These are identified using the preprocessing techniques discussed in Sec. 5.1.

Possible transitions for each monolingual HMM are $l_i \rightarrow l_i$, $l_i \rightarrow xl_i$ and $xl_i \rightarrow l_i$. We do not have the $xl_i \rightarrow xl_i$ transition, because preprocessing (Sec. 5.1) concatenates successive *universal tokens* into a single token. This does not change the output as the tokens can easily be separated after LD, but is a useful simplification for the model. The transition values for l_i are initialized by the probability of transitions between words and *universal tokens* in the text from $\mathcal{W}$.

As stated earlier, the model supports code-switching by the addition of transitions $l_i \rightarrow l_j$, and $xl_i \rightarrow l_j$, for all $i \neq j$. For each state l_i, there are $2k - 2$ new transitions (Fig. 2). We initialize these news edges with a small probability π, before normalizing transitions for each state. π, which we call the code-switch probability, is a hyperparameter tuned on a validation set.

Starting with the initialized matrices, we reestimate the transition and emission matrices using the EM-like *Baum-Welch* algorithm (Welch, 2003) over the large set of unlabeled text $\mathcal{U}$.

4.2 Decoding

The input to the trained model is first preprocessed as described in Sec. 5.1 (tokenization and identification of *universal tokens*). The Viterbi algorithm is then used with the HMM parameters to perform word-level LD. When an unknown n-gram, is encountered, its emission probability is estimated by recursively backing off to $(n-1)$-gram, until we find a known n-gram. If the unigram, i.e., the token, is also unknown, then the observation of the symbol unk is used instead.

5 Dataset Creation

The data for both training and testing comes primarily from Twitter because of its public API, and studies have shown the presence of code-switching in social media (Crystal, 2001; Herring, 2003; Danet and Herring, 2007; Cardenas-Claros and Isharyanti, 2009; Bali et al., 2014).

Our experiments use monolingual and code-switched tweets in seven languages – Dutch (*nl*), English (*en*), French (*fr*), German (*de*), Portuguese (*pt*), Spanish (*es*) and Turkish (*tr*). These form the set $\mathcal{L}$. The choice of languages is motivated by several factors. First, LD is non-trivial as all these languages use the Latin script. Second, a large volume of tweets are generated in these languages.

Third, there is annotated code-switched data available in *nl-tr* and *en-es*, which can be used for validation and testing. Lastly, we know that certain pairs of these languages are code-switched often.

5.1 Collection and Preprocessing

Using the Twitter API (Twitter, 2013), we collected tweets over May-July 2015. We selected tweets identified by Twitter LD API (Twitter, 2015) as one of the languages in $\mathcal{L}$. We also removed non-Latin script tweets.

As preprocessing, each tweet is first tokenized using *ark-twitter* (Gimpel et al., 2011) and URLs, hashtags and user mentions are identified using regular expressions. We also identify emoticons, punctuation, digits, special characters, and some universal interjections and abbreviations (such as `RT`, `aww`) as *universal tokens*. We use an existing dictionary (Chittaranjan et al., 2014) for the latter. Let the set of tweets after preprocessing be $\mathcal{T}$.

5.2 Sets $\mathcal{W}$ and $\mathcal{U}$

We use the CoverSet algorithm (Gella et al., 2014) on each tweet in $\mathcal{T}$. It obtains a confidence score for a word w_i belonging to a language l_j using a Naive Bayes classifier trained on Wikipedia. These scores are used to find the minimal set of languages are required to label all the input words. If CoverSet detects the tweet as monolingual (i.e., one language can label all words) and the identified language is the same as the Twitter LD label, the tweet is added to the weakly-labeled set $\mathcal{W}$. These tweets are almost certainly monolingual, as CoverSet has very high recall (and low precision) for detecting code-switching. As these are not manually labeled, we call them *weakly-labeled*. $\mathcal{W}$ contains 100K tweets in each language (700K in total).

From $\mathcal{T}$, we randomly select 100K tweets in each of the seven languages based on the Twitter LD API labels. These tweets do not have word-level language labels and may be code-switched or have an incorrect Twitter language label. We use these as unlabeled data, the set $\mathcal{U}$.

5.3 Validation and Test Sets

We curate two word-level gold-standard datasets for validation and testing. These sets contain monolingual tweets in each of the seven languages as well as code-switched tweets from certain language pairs, based on the availability of real-world data. However, it must be noted that GWLD can detect code-switching between more than two languages. The language-wise distribution is shown in Table 1. Including *universal tokens*, the validation and test set contain 33981 and 58221 tokens respectively. The annotated tweets will be made available for public use.

L1-L2	Tweets	L1 Tokens	L2 Tokens
nl	100 (100)	965 (1099)	–
fr	100 (102)	1085 (1045)	–
pt	100 (100)	1080 (967)	–
de	101 (100)	1078 (890)	–
tr	100 (100)	939 (879)	–
es	100 (100)	1067 (1119)	–
en	100 (100)	1161 (1006)	–
nl-en	65 (50)	498 (436)	243 (174)
fr-en	50 (48)	428 (370)	224 (227)
pt-en	53 (53)	463 (513)	278 (242)
de-en	49 (50)	417 (459)	293 (292)
tr-en	50 (50)	347 (336)	238 (209)
es-en	3013 (52)	8510 (355)	16356 (395)
nl-tr	735 (728)	5895 (8590)	5293 (8140)

Table 1: Test Set Statistics (Validation Set in parentheses). Rows in gray show existing datasets.

For *es-en*, we use the word-level annotated test set from the code-switching shared task on language detection (Solorio et al., 2014). We ignore the tokens labeled NE, Ambiguous and Mixed during our system evaluation (Sec. 6), as they do not fall in the scope of this work. The words labeled 'Other' were marked as xl_i where l_i is *en* or *es*, based on the context. We also use existing *nl-tr* validation and test sets (Nguyen and Dogruöz, 2013), which contain posts from a web forum.

For the other language pairs, we created our own validation and test sets, as none already exist. We randomly selected tweets for which CoverSet identified code-switching with high confidence. We gave 215 of these to six annotators for word-level annotation. It is difficult to find annotators who know all seven languages; elaborate guidelines were provided on using online machine translation, dictionaries and search engines for the task. Four out of the six annotators had high inter-annotator agreement – the agreement on $L1$ (language that the majority of the words in the tweet belong to) was 0.93, $L2$ (the other language, whenever present) was 0.8 and whether the tweet is code-switched was 0.84. We did not find any instances of code-switching between more than two

Systems	*Acc*	L_1L_2Acc	*IsMix*
Dictionary-based Baselines			
MAXFREQ	0.824	0.752	0.600
MINCOVER	0.853	0.818	0.733
Existing Systems			
LINGUINI	NA	0.529	0.783
LANGID	NA	0.830	0.783
POLYGLOT	NA	0.521	0.692
GWLD: The Proposed Method			
Initial	0.838	0.825	0.837
Reestimated	**0.963**	**0.914**	**0.88**

Table 2: Performance of LD Systems on Test Set

languages, which is rare in general. We distributed 3000 tweets between the four annotators (monolingual and code-switched tweets from COVERSET). Disagreements were settled between the annotators and a linguist. A subset of the annotated tweets form the validation and test sets (Table 1), and were removed from $\mathcal{W}$ and $\mathcal{U}$.

6 Experiments and Results

We compare GWLD with three existing systems: LINGUINI (Prager, 1999), LANGID (Lui and Baldwin, 2012), and POLYGLOT (Lui et al., 2014). None of these perform word-level LD, however, LANGID and POLYGLOT return a list of languages with confidence scores for the input. Since code-switching with more than two languages is absent in our dataset, we consider up to two language labels. We define the tweet to be monolingual if the difference between the confidence values for the top two languages is greater than a parameter δ. Otherwise, it is assumed to be code-switched with the top two languages. δ is tuned independently for the two LD systems on the validation set by maximizing the metric L_1L_2 *Accuracy* (Sec. 6.2). Inspired by Gella et al. (2013), we also compare with dictionary-based word-level LD baselines.

6.1 Dictionary-based Baselines

For each language, we build a lexicon of all the words and their frequencies found in $\mathcal{W}$ for that language. Let the lexicon for language $l_i \in \mathcal{L}$ be lex_i. Let $f(lex_i, w_j)$ be the frequency of w_j in lex_i. We define the following baselines:

MAXFREQ: For each w_j in $\mathbf{w}$, MAXFREQ returns lex_i that has the maximum frequency for that token. Therefore, the language label for w_j is $y_j = l_{[\arg\max_i f(lex_i, w_j)]}$. If the token is not found in any lexicon, y_j is assigned the value of y_{j-1}.

MINCOVER: We find the smallest subset $mincov(\mathbf{w}) \subset \mathcal{L}$, such that for all w_j in input $\mathbf{w}$, we have at least one language $l_i \in mincov(\mathbf{w})$ with $f(lex_i, w_j) > 0$. If there is no such language, then w_j is not considered while computing $mincov(\mathbf{w})$. Once $mincov(\mathbf{w})$ is obtained, labels y_i are computed using the MAXFREQ strategy, where the set of languages is restricted to $mincov(\mathbf{w})$ instead of $\mathcal{L}$. Note that $mincov(\mathbf{w})$ need not be unique for $\mathbf{w}$; in such cases, we choose the $mincov(\mathbf{w})$ which maximizes the sum of lexical frequencies based on MAXFREQ labels.

6.2 Metrics

We define the *Accuracy* (*Acc*) of an LD system as the fraction of words in the test set that are labeled correctly. Since the existing LD systems do not label languages at word-level, we also define:

IsMix is the fraction of tweets that are correctly identified as either monolingual or code-mixed.

L_1L_2 *Accuracy* (L_1L_2Acc) is the mean accuracy of detecting language(s) at tweet-level. For monolingual tweets, this accuracy is 1 if the gold standard label is detected by the LD system, else 0. For code-switched tweets, the accuracy is 1 if both languages are detected, 0.5 if one language is detected, and 0 otherwise. L_1L_2Acc is the average over all test set tweets.

6.3 Results

We use these metrics to assess performance on the test set for the baselines, existing LD systems and GWLD (Table 2). *Initial* refers to the HMM model estimated from $\mathcal{W}$ and *Reestimated* refers to the final model after Baum-Welch reestimation. The parameter π is tuned on the validation set using grid search. *Reestimated* GWLD has the best accuracy of 0.963 and performs significantly better than all the other systems for all metrics. Reesimatation improves the word-level *Acc* for L1 from 0.89 to 0.97 and for L2 from 0.43 to 0.82. LINGUINI and POLYGLOT likely have low L_1L_2Acc because they are trained on synthetically-created documents with no word-level code-switching.

Since our test set contains pre-existing annotations for *en-es* (Solorio et al., 2014) and *nl-tr* (Nguyen and Dogruöz, 2013), we compare with state-of-the-art results on those datasets. On *en-es* tokens, Al-Badrashiny and Diab (2016) reports an $F1$-score of 0.964; GWLD obtains 0.978. Nguyen and Dogruöz (2013) report 0.976 *Acc* on the *nl-tr*

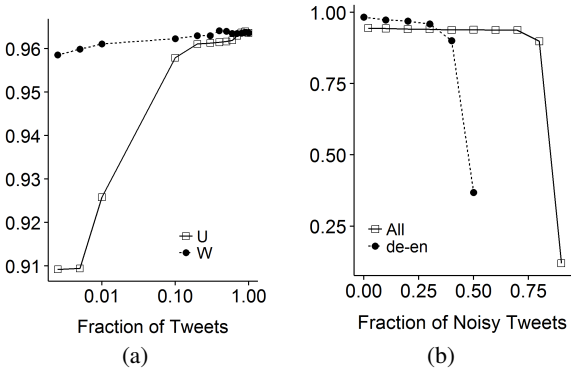

Figure 3: Acc versus Dataset Parameters

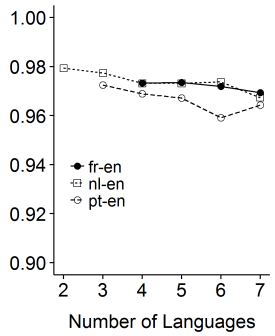

Figure 4: Acc versus Number of Languages

test set. We obtain a less competitive 0.936. However, when errors between *nl-en* are ignored as most of these are *en* words with *nl* gold-standard labels (convention followed by the dataset creators), the revised Acc is 0.963. Notably, unlike GWLD, both these models use large amounts of annotated data for training and are restricted to detecting only two languages.

Error Analysis: GWLD sometimes detects languages that are not present in the tweet, which account for a sizable fraction (39%) of all word-level errors. Not detecting a language switch causes 8% of the errors. Most other errors are caused by named entities, single-letter tokens, unseen words and the *nl-en* annotation convention in the test set from Nguyen and Dogruöz (2013).

6.4 Robustness of GWLD

We test the robustness of GWLD by varying the size of the weakly-labeled set, the unlabeled dataset and the number of languages the model is trained to support.

6.4.1 Size of $\mathcal{W}$ and $\mathcal{U}$

The variation of Acc with the size of $\mathcal{W}$ is shown in Figure 3a. Even with 0.25% of the set (250 tweets for each $l_i \in \mathcal{L}$), the model has accuracy of nearly 0.96. A slow rise in accuracy is observed as the number of tweets in $\mathcal{W}$ is increased. We also experiment with varying the size of $\mathcal{U}$. In Figure 3a, we see that with 0.25% of U (around 1,400 randomly sampled tweets), the accuracy on the test set is lower than 0.91. This quickly increases with 10% of $\mathcal{U}$. Thus, GWLD achieves Acc comparable to existing systems with very little weakly-labeled data (just 250 tweets per language, which are easily procurable for most languages) and around 50,000 unlabeled tweets.

L1-L2	Acc	$IsCM$	GWLD-Acc
nl-en	0.979	0.943	0.967
fr-en	0.982	0.948	0.969
pt-en	0.977	0.952	0.964
de-en	0.984	0.956	0.975
tr-en	0.985	0.984	0.983
es-en	0.954	0.929	0.978
nl-tr	0.975	0.907	0.936

Table 3: Statistics for Pairwise (col. 2 and 3) and GWLD Systems

6.4.2 Noise in $\mathcal{W}$

Since a small, but pure, $\mathcal{W}$ gives high accuracy (Sec. 6.4.1), we evaluate how artificially introduced noise affects Acc. The noise introduced into the $\mathcal{W}$ of each language comes uniformly from the other six languages. Figure 3b shows how increasing fractions of noise slowly degrades accuracy, with a steep drop to 0.11 accuracy at 90% noise, where the tweets from each incorrect language outnumber the correct language tweets. We test this with a pairwise model as well, as noise from a single language might have greater effect. The accuracy falls to 0.36 at 50% noise (Fig. 3b). At this point, $\mathcal{W}$ has an equal number of tweets from each language and is essentially useless.

6.4.3 Number of languages

Pairwise Models: Table 3 details two performance metrics (defined in Sec. 5.2) for our model trained on only two languages and the corresponding 7-language GWLD Acc for that language pair.

Incremental Addition of Languages: We test Acc while incrementally adding languages to the model in a random order (*nl-en-pt-fr-de-es-tr*). Figure 4 shows the variation in Acc for *nl-en*, *pt-en* and *fr-en* as more languages are added to the

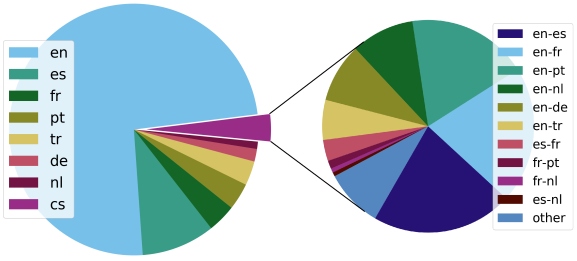

Figure 5: Worldwide distribution of monolingual and CS tweets (left and right charts respectively)

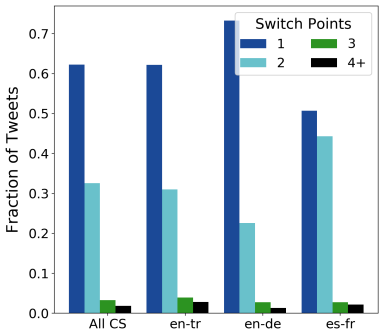

Figure 6: Worldwide CS point distribution

model. Although there is a slight degradation, in absolute terms, the accuracy remains very high.

7 Code-Switching on Twitter

The high accuracy and fast processing speed (the current multithreaded implementation labels 2.5M tweets per hour) of GWLD enables us to conduct large-scale and reliable studies of CS patterns on Twitter for the 7 languages. In this paper, we conduct two such studies. The first study analyzes 50M tweets from across the world to understand the extent and broad patterns of switching among these languages. In the second study, we analyze 8M tweets from 24 cities to gain insights into geography-specific CS patterns.

7.1 Worldwide Code-Switching Trends

We collected 50 million unique tweets that were identified by the Twitter LD API as one of the 7 languages. We place this constraint to avoid tweets from unsupported languages during analysis. Figure 5 shows the overall language distribution, including the CS language-pair distribution. Approximately 96.5% of the tweets are monolingual, a majority of which are *en* (74%).

Around 3.5% of all tweets are code-switched. Globally, *en-es*, *en-fr* and *en-pt* are the three most commonly mixed pairs accounting for 21.5%, 20.8% and 18.4% of all CS tweets in our data respectively. Interestingly, 85.4% of the CS tweets have *en* as one of the languages; *fr* is the next most popularly mixed language, with *fr-es* (3.2%), *fr-pt* (1.2%) and *fr-nl* (0.6%) as the top three observed pairs. Although around 1% of CS tweets were detected as containing more than two languages, these likely have low precision because of language overdetection as discussed in Sec. 6.3.

Figure 6 shows the fraction of *code-switch points*, i.e., how many times the language changes in a CS tweet, for all the languages, as well as for three language pairs with to highlight different trends. Most CS tweets have one CS-point, which implies that the tweet begins with one language, and then ends with another. Such tweets are very frequent for *en-de* where we observe that usually the tweets state the same fact in both *en* and *de*. This so-called *translation function* (Begum et al., 2016) of CS is probably adopted for reaching out to a wider and global audience. In contrast, *es-fr* tweets have fewer tweets with single and far more with two CS-point than average. Tweets with two CS-points typically imply the inclusion of a short phrase or chunk from another language. *en-tr* tweets have the highest number of CS-points, implying rampant and fluid switching between the two languages at all structural levels.

7.2 City-Specific Code-Switching Trends

Cosmopolitan cities are melting pots of cultures, which make them excellent locations for studying multilingualism and language interaction, including CS (Bailey et al., 2013). We collected tweets from 24 populous and highly cosmopolitan cities from Europe, North America and South America, where the primarily spoken language is one of the 7 languages detectable by GWLD. Around 8M tweets were collected from these cities.

Table 4 shows the top and bottom 6 cities, ranked by the fraction of CS tweets from that city. The total number of tweets analyzed and the top two CS pairs, along with their fractions (of CS tweets from that city) are also reported. More details can be found in the supplementary material. It is interesting to note that the 6 cities with lowest CS tweet fractions have *en* as the major language, whereas the 6 cities with highest CS fractions are from non-English (Turkish, Spanish and French) speaking geographies. In fact, the Pearson's cor-

Cities with highest fraction of CS tweet			Cities with lowest fraction of CS tweets		
City	Tweets	CS-fraction (CS pairs)	City	Tweets	CS-fraction (CS pairs)
Istanbul	351K	.12 (*en-tr* .53, *nl-tr* .13)	Houston	588K	.01 (*en-es* .22, *en-fr* .21)
Québec City	108K	.08 (*en-fr* .45, *es-fr* .23)	San Francisco	532K	.02 (*en-es* .26, *en-fr* .19)
Paris	158K	.07 (*en-fr* .43, *fr-pt* .21)	NYC	690K	.02 (*en-es* .21, *en-fr* .19)
Mexico City	332K	.07 (*en-es* .54, *es-fr* .14)	Miami	290K	.02 (*en-es* .33, *en-pt* .20)
Brussels	100K	.06 (*en-fr* .37, *es-fr* .15)	London	492K	.02 (*en-fr* .26, *en-pt* .17)
Madrid	147K	.06 (*en-es* .43, *es-fr* .32)	San Diego	432K	.02 (*en-es* .29, *en-fr* .14)

Table 4: Top (left) and bottom (right) six cities according to the fraction of CS tweets.

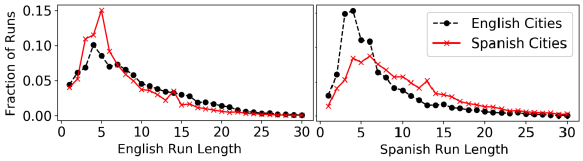

Figure 7: *en-es* Run Length

relation between the fraction of monolingual English tweets and CS tweets for these 24 cities is -0.85. Further, from Table 4 one can also observe that for non-English speaking geographies, the majority language is most commonly mixed with English, followed by French (Spanish, if French is the majority language). Istanbul is an exception, where Dutch is the second most commonly mixed language with Turkish, presumably because of the large Turkish immigrant population in Netherlands resulting in a sizeable Turkish-Dutch bilingual diaspora (Doğruöz and Backus, 2009; Nguyen and Dogruöz, 2013).

Is there a difference in the way speakers mix a pair of languages, say *en* and *es*, in *en*-speaking goegraphies like San Diego, Miami, Houston and New York City, and *es*-speaking geographies like Madrid, Barcelona, Buenos Aires and Mexico City? Indeed, as shown in Fig. 7, the distribution of the lengths of *en* and *es* runs (contiguous sequence of words in a single language beginning and ending with either a CS-point or beginning/end of a tweet) in *en-es* CS tweets is significantly different in *en*-speaking and *es*-speaking geographies. *en* runs are longer in *en*-speaking cities and vice versa, showing that the second language is likely used in short phrases.

8 Conclusion and Future Work

We present GWLD, a system for word-level language detection for an arbitrarily large set of languages that is completely unsupervised. Our results on monolingual and code-switched tweets in seven Latin script languages show a high 0.963 accuracy, significantly out-performing existing systems. Using GWLD, we conducted a large-scale study of CS trends among these languages, both globally and in specific cities.

One of the primary observations of this study is that while code-switching on Twitter is common worldwide (3.5%), it is much more common in non-English speaking cities like Istanbul (12%) where 90% of the population speak Turkish. On the other hand, while a third of the population of Houston speaks Spanish and almost everybody English, only 1% of the tweets from the city are code-switched. All the trends indicate a global dominance of English, which might be because Twitter is primarily a medium for broadcast, and English tweets have a wider audience. Bergsma et al. (2012) show that "[On Twitter] bilinguals bridge between monolinguals with English as a hub, while monolinguals tend not to directly follow each other." Androutsopoulos (2006) argues that due to linguistic non-homogenity of online public spaces, languages like *en*, *fr* and *de* are typically preferred for communication, even though in private spaces, "bilingual talk" differs considerably in terms of distribution and CS patterns.

As future directions, we plan to extend GWLD to several other languages and conduct similar sociolinguistic studies on CS patterns including not only more languages and geographies, but also other aspects like topic and sentiment.

Acknowledgments

We would like to thank Prof. Shambavi Pradeep and her students from BMS College of Engineering for assisting with data annotation. We are also grateful to Ashutosh Baheti and Silvana Hartmann from Microsoft Research (Bangalore, India) for help with data organization and error analysis.

References

Mohamed Al-Badrashiny and Mona Diab. 2016. Lili: A simple language independent approach for language identification. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING). Osaka, Japan*.

Jannis Androutsopoulos. 2006. Multilingualism, diaspora, and the internet: Codes and identities on german-based diaspora websites. *Journal of Sociolinguistics* 10(4):520–547.

Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

George Bailey, Joseph Goggins, and Thomas Ingham. 2013. What can Twitter tell us about the language diversity of Greater Manchester? In *Report by Multilingual Manchester*. School of Languages, Linguistics and Cultures at the University of Manchester. http://bit.ly/2kG42Qf.

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*.

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*.

Rafiya Begum, Kalika Bali, Monojit Choudhury, Koustav Rudra, and Niloy Ganguly. 2016. Functions of code-switching in tweets: An annotation framework and some initial experiments. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific twitter collections. In *Proceedings of the second workshop on language in social media*. Association for Computational Linguistics.

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*.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal* 47:195–215.

William B Cavnar and John M Trenkle. 1994. N-gram-based text categorization .

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language* 13(4):359–393.

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*.

Monojit Choudhury, Gokul Chittaranjan, Parth Gupta, and Amitava Das. 2014. Overview of FIRE 2014 track on transliterated search .

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.

Amitava Das and Bjorn 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*. Goa, India, pages 169–178.

A Seza Doğruöz and Ad Backus. 2009. Innovative constructions in dutch turkish: An assessment of ongoing contact-induced change. *Bilingualism: language and cognition* 12(01):41–63.

Ted Dunning. 1994. *Statistical identification of language*. Computing Research Laboratory, New Mexico State University.

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2013. Code switch point detection in arabic. In *Natural Language Processing and Information Systems*, Springer, pages 412–416.

Archana Garg, Vishal Gupta, and Manish Jindal. 2014. A survey of language identification techniques and applications. *Journal of Emerging Technologies in Web Intelligence* 6(4):388–400.

Spandana Gella, Kalika Bali, and Monojit Choudhury. 2014. "ye word kis lang ka hai bhai?" testing the limits of word level language identification. In *NLPAI*.

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

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and A. Noah Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*.

Moises Goldszmidt, Marc Najork, and Stelios Paparizos. 2013. Boot-strapping language identifiers for short colloquial postings. In *Machine Learning and Knowledge Discovery in Databases*, volume 8189 of *Lecture Notes in Computer Science*, pages 95–111.

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

Harald Hammarström. 2007. A fine-grained model for language identification. In *In Workshop of Improving Non English Web Searching. Proceedings of iNEWS 2007 Workshop at SIGIR*.

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

Baden Hughes, Timothy Baldwin, SG Bird, Jeremy Nicholson, and Andrew MacKinlay. 2006. Reconsidering language identification for written language resources .

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada.

Suin Kim, Ingmar Weber, Li Wei, and Alice Oh. 2014. Sociolinguistic analysis of twitter in multilingual societies. In *Proceedings of the 25th ACM conference on Hypertext and social media*.

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

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *In Proceedings of the ACL 2012 System Demonstrations*. pages 25–30.

Marco Lui, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. In *Transactions of the Association for Computational Linguistics*.

Ed Manley. 2012. Detecting languages in Londons Twittersphere. In *Blog post: Urban Movements*. http://bit.ly/2kBytHm.

P. McNamee. 2005. Language identification: A solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges* 20.

Lesley Milroy and Pieter Muysken. 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press.

Giovanni Molina, Nicolas Rey-Villamizar, Thamar Solorio, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, and Mona Diab. 2016. Overview for the second shared task on language identification in code-switched data. *EMNLP 2016* page 40.

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

Dong Nguyen and A. Seza Dogruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Dong Nguyen, A Seza Doğruöz, Carolyn P Rosé, and Franciska de Jong. 2016. Computational sociolinguistics: A survey. *Computational Linguistics* .

Shana Poplack. 1980. Sometimes Ill start a sentence in Spanish y termino en espaol. *Linguistics* 18:581–618.

John M Prager. 1999. Language identification for multilingual documents. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference*.

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

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do Hindi-English speakers do on Twitter? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

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

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

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *In Proc. 20th Machine Learning conference of Belgium and The Netherlands*. pages 27–34.

Twitter. 2013. GET statuses/sample — Twitter Developers. https://dev.twitter.com/docs/api/1/get/statuses/sample.

Twitter. 2015. GET help/languages — Twitter Developers. https://dev.twitter.com/rest/reference/get/help/languages.

Lloyd R Welch. 2003. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter* 53(4):10–13.

Fei Xia, William D Lewis, and Hoifung Poon. 2009. Language id in the context of harvesting language data off the web. In *In Proceedings of the 12th EACL*. pages 870–878.

Wei Zhang, Robert AJ Clark, Yongyuan Wang, and Wen Li. 2016. Unsupervised language identification based on latent dirichlet allocation. *Computer Speech & Language* 39:47–66.