Collecting Code-Switched Data from Social Media

Gideon Mendels†, Victor Soto†, Aaron Jaech, Julia Hirschberg
Columbia University, New York, NY 10027
University of Washington, Seattle, WA 98105
g.mendels@columbia.edu, vsoto@cs.columbia.edu, ajaech@uw.edu, julia@cs.columbia.edu

Abstract

We address the problem of mining code-switched data from the web, where code-switching is defined as the tendency of bilinguals to switch between their multiple languages both across and within utterances. We propose a method that identifies data as code-switched in languages L_1 and L_2 when a language classifier labels the document as language L_1 but the document also contains words that can only belong to L_2. We apply our method to Twitter data and collect a set of more than 43,000 tweets. We obtain language identifiers for a subset of 8,000 tweets using crowd-sourcing with high inter-annotator agreement and accuracy. We validate our Twitter corpus by comparing it to the Spanish-English corpus of code-switched tweets collected for the EMNLP 2016 Shared Task for Language Identification, in terms of code-switching rates, language composition and amount of code-switch types found in both datasets. We then trained language taggers on both corpora and show that a tagger trained on the EMNLP corpus exhibits a considerable drop in accuracy when tested on the new corpus and a tagger trained on our new corpus achieves very high accuracy when tested on both corpora.

Keywords: code-switching, data collection, social media

1. Introduction

Linguistic code-switching is the phenomenon by which bilingual speakers switch back and forth between languages during communication. Code-switching can be classified as inter-sentential when the switch occurs between the boundaries of a sentence or utterance, or intra-sentential when it occurs within those boundaries. For example a Spanish-English speaker might say "Me dijo que Juanito is very good at math," which represents an intra-sentential switch, or "Me dijo que Juanito mintió. I don't believe it!"

Code-switching can be observed at various linguistic levels of representation for different language pairs: phonological, morphological, lexical, syntactic, semantic, and discourse/pragmatic switching. However, very little code-switching corpora exist from which researchers can train statistical models. The question of how to acquire code-switched data from web and social media resources automatically and accurately remains largely unaddressed.

In this paper, we present a method to automatically collect code-switched data from Twitter. Twitter data has been mined extensively for many Natural Language Processing and speech tasks (Mendels et al., 2015; Go et al., 2009) as one of the only major platforms that provides an API for data collection. The proposed method, which we term "anchoring" can also be used for collecting data from other sources.

The remainder of the paper is organized as follows. In Section 2. we give an overview of previous work on the topic of finding and collecting code-switched data. In Section 3. we present our anchoring method for retrieving code-switched tweets. Section 4. provides the details of our Twitter collection pipeline. Section 5. describes the language identification (LID) task we used to crowdsource the word language tags for the data collected. In Section 6.1., we compare the corpus we acquired using this method with a corpus of tweets that was collected for the EMNLP 2016 Shared Task for Language Identification in code-switched (CS) Data. We compare them in terms of the amount of bilingualism they contain and their code-switching rate – i.e., how frequently writers switch their language in the corpus. In Section 6.2. we train and test language ID taggers on our corpus and the Workshop corpus and compare their performance. We present our conclusions in Section 7.

2. Previous Work

In the past few years there have been increasing efforts on a variety of tasks using code-switched data, including part-of-speech tagging (Solorio and Liu, 2008b; Vyas et al., 2014; Jamatia et al., 2015; AlGhamdi et al., 2016), parsing (Goyal et al., 2003), language modeling (Franco and Solorio, 2007; Li and Fung, 2012; Adel et al., 2013b; Adel et al., 2013a; Li and Fung, 2014), code-switching prediction (Solorio and Liu, 2008a; Elfardy et al., 2014), sentiment analysis (Vilares et al., 2015; Lee and Wang, 2015) and even speech recognition (Ahmed and Tan, 2012; Lyudovyk and Pylypenko, 2014).

The task that has received most of the attention has been Language Identification on code-switched data, thanks in part to the First and Second Shared Tasks on EMNLP 2014 and 2016 (Solorio et al., 2014; Molina et al., 2016). Many of the current state-of-the-art models for Language Identification perform sequence labeling using Conditional Random Fields (Al-Badrashiny and Diab, 2016) or Recurrent Neural Networks (Jaech et al., 2016b). In the 2016 Shared Task the best performing system on the MSA-DA dataset used a combination of both (Samih et al., 2016) on top of word and character-level embeddings, and the best performing system on the ES-EN dataset used logistic regression (Piergallini et al., 2016) and character n-gram features.

On the task of finding and collecting code-switched data from the web, which is the focus of this paper, Çetinoglu (2016) obtained a corpus of German-Turkish tweets by automatically computing dictionaries of pure German and Turkish from a million Turkish, German and English

†The first two authors contributed equally to this work.

tweets. They subsequently used those dictionaries to automatically tag ten million Turkish tweets from which they obtained 8,000 potentially code-switched tweets which they manually filtered down to 680.

Samih (2016) obtained a corpus of forum posts written in MSA and the Darija Dialect following this iterative process: they first started with a list of 439 words exclusive to Darija, they retrieved forum posts that contained one of the exclusive words, and then they added all the words from the retrieved posts to the list of Darija words. They repeated the process until the corpus reached a certain size. Even though their method offers no guarantees, they obtained a corpus with 73.9% of code-switched forum posts.

Barman et al. (2014) used a group of university students as data source to find code-switched media. They found a Facebook group and 11 Facebook users from which they collected 2,335 posts and 9,813 comments. Vyas et al. (2014) collected almost seven thousand comments from 40 manually selected code-switched Facebook posts from three celebrity pages and the BBC Hindi news page. Finally, Jamatia et al. (2015) collected tweets and Facebook posts from a University billboard page, although it is unclear if they specifically targeted code-switched content or not.

The organizers of the EMNLP Shared Tasks on Language Identification in code-switched Data followed a semi-automatic approach. For the first Shared task, code-switched data was collected for the pairs Spanish-English (ES-EN), Mandarin-English (MAN-EN), Nepali-English (NEP-EN) and Modern Standard Arabic-Dialectal Arabic (MSA-DA). The social media sources they targeted were Twitter for all language pairs and Facebook for NEP-EN and blog comments for MSA-DA. For Twitter, their approach consisted in first locating code-switchers and then collecting their posts and posts from their followers and/or followees. For ES-EN, they located a subset of code-switchers by querying the Twitter API with frequent English words, and restricted results to tweets identified as Spanish by Twitter from users based in Texas and California. For NEP-EN, they started from a group of acquaintances that were known to code-switch and then identified their followers and followers of their followers that they found were code-switchers too. For Mandarin-English, they started by looking at the most followed Twitter users in Taiwan. They then added those users that they manually checked were code-switchers to their pool, and repeated a similar process on their followees. For MSA-DA, they seeded the search with text from Egyptian public figures. For the Second Shared task the language pairs were ES-EN and MSA-DA. For ES-EN they restricted the search of code-switchers based in New York and Miami and seeded the search from local radio station accounts. Again, they continued looking for followers and followees of the radio stations that tweeted code-switched messages. For MSA-DA, the same collection method from the 2014 Shared Task was reused.

All of these approaches to code-switched data collection, except (Samih, 2016), rely on manual inspection to some degree in order to either add a user to the code-switcher pool or select a post for collection. In the next section we introduce a fully automatic approach to finding and collecting code-switched data that is not dependent on manually curating lists of users.

3. Anchoring Methods

We define an anchor as a word which belongs to only one language from a large pool of languages. The motivation behind using anchor words stems from a simple rule to detecting code-switched sentences: "A sentence is code-switched in $L_1 + L_2$ if and only if it contains at least one anchor from language L_1 and at least one anchor from language L_2, and contains no anchors from any other language from the pool of languages $\mathbb{L}$."

The set of anchor words for a language L_i is computed as the set difference between its word lexicon $V(L_i)$ and the union of all other lexicons in the language pool:

$$AnchorSet(L_i) = V(L_i) \setminus \cup_{j!=i} V(L_j) \quad (1)$$

Note that the identification of the anchor sets for a given language pair depends upon the monolingual corpora used. We can relax the definition of anchors in two different ways. First, in the context of detecting $L_1 + L_2$ language, we say a word is a "weak anchor" if it is seen in monolingual L_1 corpora, and never seen in monolingual L_2 corpora. Second, querying the Twitter API with every possible pair of one Spanish and one English anchor is unproductive because there are billions of possible queries and most of them would have no results. To avoid this problem we relaxed the definition of code-switching to: "a sentence is code-switched if and only if it is predicted to be L_1 by a monolingual automatic Language Identification program and contains at least one weak anchor from the L_2 anchor set." With this new rule we require only one anchor from one of our language pair plus language id results favoring the other member of the pair. We note that the definition of weak anchors closely resembles the definition of blacklisted words used by (Tiedemann and Ljubešić, 2012), although their application was to discriminate between a set of very similar languages (Serbian, Croatian and Bosnian). Using these definitions, we performed a preliminary study on the task of classifying an utterance as monolingual or code-switched on the EMNLP 2016 Shared Task Corpus of Spanish+English tweets. Details of the collection and contents of that corpus were given in Section 2.. We computed the anchors for Spanish and English from the Leipzig corpora Collection (LCC), released 2007 to 2014 (Goldhahn et al., 2012). The LCC is a collection of corpora for a large set of languages from comparable sources (e.g. Wikipedia, news articles, websites). We computed the word lexicon of every language in the corpus from the news dataset for that language, and then we computed the anchor list first following equation 1. Words that contained numbers or tokens from a list of 31 punctuation tokens were discarded. In total the language pool contained 134 languages. The Spanish anchor set contained 50.68% of the words from the Spanish word lexicon and the English anchor set contained 54.37% of the words from the English lexicon. In both cases, this is one of the smaller percentages from the pool of 134 languages. In comparison, German, French

Method	Class	Prec.	Recall	F1-score
Anchors	Mono	0.58	1.00	0.73
	CS	0.94	0.03	0.07
Weak Anchors	Mono	0.68	0.98	0.80
	CS	0.93	0.38	0.54
Weak +LID	Mono	0.66	0.98	0.79
	CS	0.93	0.33	0.49

Table 1: Performance on the task of code-switched sentence detection using three definitions for anchoring.

and Italian kept 79.01, 59.67 and 62.94% of their lexicons, while other languages like Chinese and Japanese kept 93.40 and 72.18%.

Table 1 shows Precision, Recall and F1-Score results on the task of classifying a tweet as code-switched (CS) or monolingual (mono) for the strong definition of anchors, weak anchors and the weak anchor + LID approach. We report results on the test partition of the EMNLP 2016 Shared Task Corpus. The language ID used is langid.py (Lui and Baldwin, 2012).

The top subtable from Table 1 shows the results we obtained for this task using our strong definition of anchor. Not surprisingly, we achieved very high precision, but very low recall. High precision and low recall is a secondary effect from the restrictiveness of the definition of anchor set and code-switched sentence, since anchors are not defined exclusively in terms of L_1 and L_2, but from a large pool of languages. This means that the words in the anchor set are most likely to be very low-frequency words. Furthermore the fact that a sentence must have at least one anchor from both languages and none from all the other languages, guarantees that much of the data will be rejected as not code-switched even when bilingual speakers of the languages in question would agree that it is.

The middle subtable from Table 1 shows the results on the task using weak anchors as defined above. At the expense of 0.01 absolute precision points, recall is improved by almost 0.35 points.

The bottom subtable of Table 1 shows results using weak anchors and Language Id. Although with this method the recall drops 0.03 points with respect to the weak anchors, we achieve the advantage of being able to reduce the number of queries we need for the collection, and make the search less restrictive. In the next section of the paper we use weak anchors with the Language ID restriction to collect code-switched tweets.

4. Data Collection

We used Babler[1] (Mendels et al., 2016) to collect code-switched data from Twitter. Babler is a tool designed for harvesting web-data for NLP and machine learning tasks. Babler's pipeline is launched by querying a seed word $s \in S$ using Twitter's API. The tweets retrieved by the query are later processed and passed through a set of filtering rules R which are predefined for the task.

Following the definition of "weak anchor plus Language Id" given in section 3. we used the "weak" anchors to seed the Twitter API and the filtering rules R to enforce the LID restriction. To further reduce the number of required queries we also sort our "weak" anchors by frequency. The weak anchors were computed from the GigaCorpus dataset of Broadcast News data (Consortium and others, 2003; Graff, 2011). R uses Twitter's LID to only allow tweets that were seeded from a Spanish anchor and classified as English or vice versa. Although we require the Twitter API to return only exact matches to our seed terms, we found that in fact Twitter performs stemming.

Our method differs from the prior art in two aspects. First, we derive our word lists from non-noisy pure monolingual corpora which reduces the risk of including out-of-language tokens. Second, instead of performing local filtration our method is implemented based only on API calls thus increasing our potential dataset to every public tweet available. Overall we collected 14,247 tweets that were seeded from Spanish weak anchors and classified as English by the Twitter API and 28,988 tweets that were seeded from English weak anchors and classified as Spanish.

5. Crowdsourcing Language Tags

While we designed our data collection pipeline to save only code-switched tweets, we next needed to test this, as well as to obtain manual annotations for our language modeling research.

From the more than forty-three thousand tweets that were collected, we randomly chose a subset of 8,285 tweets for our "Anchored" tweets corpus [2]. We crowdsourced language tags for every word in our Anchored tweet dataset. Each word was tagged as English (EN), Spanish (ES), Ambiguous between English and Spanish (AMBIG), Mixed English-Spanish (MIXED), Named Entity (NE), Foreign Word (FW), Other (OTHER) and Gibberish (UNK). "Named Entities" were defined as single proper names or part of a name or title that refer to persons, places, organizations, locations, brands, goods, initials, movie titles and song titles. A word is to be tagged as "Ambiguous" when it can be used in both English and Spanish, but there is not enough context to decide its use in the current tweet. A word is to be tagged "Mixed" when the word does not exist in Spanish or English, but consists of a combination of elements from both, e.g. the word "ripeado" which contains the English root "rip" and the Spanish morpheme "-ado". The category "Other" is to be used to tag punctuation, numbers, emoticons, retweet symbols, and other non-lexical items. Finally the "Gibberish" category is for tokens whose meaning cannot be identified.

We used the guidelines used for the annotation of the EMNLP 2016 Shared Task dataset, with some minor changes, including a large number of examples per language tag, and reminders to the annotators throughout the instructions and question statements that a) hashtags were to be tagged with the language tag of the words in the hashtag, and b) Named Entities had precedence over any other

[1] Babler is publicly available from `https://github.com/gidim/Babler`

[2] All the anchor wordlists, tweet IDs and their crowdsourced language tags are publicly available in `http://www.cs.columbia.edu/~vsoto/files/lrec_2018_package.zip`

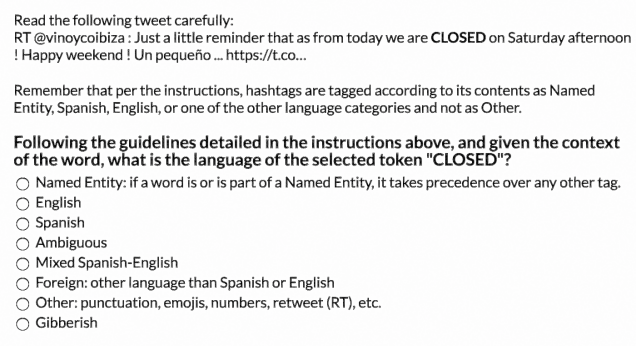

Figure 1: Instance of the annotation task implemented in Crowdflower.

Lang Tag	#Tokens	Avg. Conf
ES	40,208	0.97
EN	30,372	0.93
AMBIG	919	0.55
MIXED	129	0.54
NE	15,260	0.88
FW	1,815	0.77
OTHER	1,994	0.80
UNK	546	0.59

Table 2: Number of tokens and average confidence per Language ID tag from the crowdsourced annotations for the Anchored Twitter corpus.

language tag, since these were the test questions they had the most difficulty with in our initial test.

We used Crowdflower to crowdsource language tags for our tweets. An example of the task our workers were asked to complete can be seen in Figure 1. Our workers were prescreened using a quiz of twenty test questions. If three or more test questions were missed during the initial quiz, the worker was denied access to the task. Furthermore, workers were required to be certified for the Spanish language requirement in Crowdflower. Only workers from Argentina, Canada, Mexico, Spain, U.K. and U.S.A. were allowed access to the task. The task was designed to present 20 questions per page plus one test question used to assess workers' performance. When a worker reached an accuracy lower than 85% on these test questions, all their submitted judgments were discarded and the task made subsequently unavailable. Every set of 19+1 judgments was paid 1 cent (USD).

In total, we collected three judgments per token. The average inter-annotator agreement was 92.33% and the average test question accuracy was 91.1%. These metrics demonstrate that the crowdsourced language labels are of high-quality. For every token for which we crowdsourced a language tag, Crowdflower computes the confidence on the language tag as the level of agreement between all the contributors that predicted that language tag weighted by the contributors' trust scores. The language tag with highest confidence is then chosen as aggregated prediction. Table 2 shows the average confidence per language tag across all tokens. It can be seen that workers struggled the most when tagging words as Mixed, Ambiguous or Gibberish.

	Workshop		Anchored
	Train-Dev	Test	Full
#Tweets(K)	14.4	10.7	8.5
#Tokens(K)	172.8	121.4	130.7
#Switches(K)	7.4	7.8	10.2
Avg. #swts	0.52	0.73	**1.19**
Swt.words(%)	4.30	6.42	**7.77**
Swt.tweets(#)	4,116	4,617	5,958
Swt.tweets(%)	28.56	43.09	**69.89**
0 swt. (%)	**71.44**	56.91	30.11
1 swt. (%)	12.86	21.38	39.57
2 swt. (%)	11.34	16.65	19.53
3 swt. (%)	2.50	2.88	5.81
4 swt. (%)	1.27	1.66	3.32
5 swt. (%)	0.29	0.33	0.84
6 swt. (%)	0.20	0.17	0.43
7 swt. (%)	0.05	0.02	0.23
8 swt. (%)	0.03	0.00	0.12

Table 3: Code-switching statistics for the EMNLP 2016 Workshop and Anchored Tweets datasets. A code-switched word is a word whose language is different from the word that precedes it. The bottom subtable shows the percentage of tweets that contain *N* code-switches.

6. Evaluation

6.1. Data Assessment

Given the crowdsourced LID labels, we can assess the quality of the retrieved anchored tweets by computing their degree of bilingualism and how frequently code-switching occurs within them. We compare these measures to the EMNLP 2016 CS Shared Task corpus (Molina et al., 2016). The train and dev tweets from the 2016 Shared Task were the train and test sets from the 2014 Shared Task (Solorio et al., 2014), whereas the test split was collected specifically for the 2016 task. The collection schemes used in 2014 and 2016 were explained in detail in Section 2.. Table 3 provides the overall statistics describing this corpus in comparison to ours. We report the train-dev and test splits of the EMNLP 2016 Workshop Shared Task corpus separately since they were collected using different methods. As can be seen in Table 3, our subset of 8,525 tweets had an average of 1.19 code-switches per tweet, with 7.77% of words in a tweet being followed by a switch. 69.89% of our tweets contained at least one or more switches. In comparison, the Workshop corpus had an average of 0.61 code-switches per tweet, with 5.17% of tokens followed by a switch. Only 34.75% tweets contained at least one switch. The test set of the Workshop corpus shows greater degrees of bilingualism and a better switching rate: Test corpus tweets averaged 0.73 code-switches per tweet, with 6.42% of tokens followed by a switch and contained 43.09% code-switched tweets overall. Based on these metrics alone, it would appear that our anchoring method improves over the earlier approach considerably.

Table 4 shows the language composition of the three datasets: Workshop training-dev, Workshop test, and the full Anchored dataset. From this table we can see that the train-dev portion of the workshop corpus has a majority (>55%) of English words, while the test split contains a large majority of Spanish words (>63.44%), perhaps due to

	Workshop		Anchored
Lang Tag	Train-Dev	Test	Full
ES	24.51	63.44	34.44
EN	55.33	13.95	24.73
AMBIG	0.23	0.00	0.70
MIXED	0.04	0.00	0.10
NE	2.09	1.72	11.68
FW	0.01	0.02	1.39
OTHER	17.62	20.84	26.53
UNK	0.17	0.02	0.42

Table 4: Language composition (%, token level) for the EMNLP 2016 Workshop and Anchored Tweets datasets.

seeding the collection of tweets on Spanish-language Radio accounts and followers/ees. In comparison, the Anchored corpus is more balanced, with 34.44 and 24.73% of Spanish and English tokens. It also has a higher rate of Named Entities and Other tokens. We believe this is due to the updated annotation guidelines that emphasized the subtleties involved in annotating Named Entities and Other tokens. While Table 4 compares the corpora by language

	Workshop		Anchored
Switch Type	Train-Dev	Test	Full
ES EN	32.06	45.68	29.81
EN ES	31.47	28.36	22.86
EN Other+ ES	15.99	12.28	14.83
ES Other+ EN	15.16	11.05	10.86
ES NE+ EN	1.44	0.99	4.06
EN NE+ ES	0.91	0.36	2.45

Table 5: Types of code-switching sorted by frequency (%). TAG+ indicates a sequence of one or more occurrences of that language tag.

composition, Table 5 examines the corpora by type of switch. The most frequent switch across datasets is Spanish to English (ES-EN), followed by English to Spanish (EN-ES). These account for 63.53%, 74.04% and 52.67% of switches for the Workshop Train-Dev, Workshop Test and Anchored datasets respectively. The next most common type of switch is an English word followed by a sequence of Other tokens and a Spanish word (EN-Other-ES), or Spanish followed by Other and then English (ES-Other-EN). These make up for 31.15%, 23.33% and 25.69% of the switches. Note that this type of switch can be indicative of inter-sentential code-switching if the Other token is a punctuation mark (like '!' in "Holaaaa mis niños bellos!!! I love you guys") or it can be indicative of intra-sentential code-switching if the other token is a Twitter mention, a quote, and so on (e.g "En cuestiones de Rock 'n' Roll I am pretty crossover"). Overall, the typing distribution is more balanced in the Anchored dataset, whereas the Workshop test set has a significant majority of ES EN switches, due perhaps, again, to the way the collection of tweets was seeded.

6.2. Language Identification

Our second evaluation of the accuracy of our corpus consists of training and testing Language ID taggers on the new dataset and comparing its performance to a tagger trained on the Workshop data. We made use of a high-performing classification model from Jaech et al. (2016b). The model did well on the English-Spanish code-switching 2016 Shared Task, especially considering that it was one of only two models that did not use external resources for training (Molina et al., 2016). The same model did well on a sentence level language identification task (Jaech et al., 2016a).

We summarize the model architecture and its motivation here. For a full description see Jaech et al. (2016b). The model is a hierarchical neural model with one level that operates on character sequences to build a representation for each word and a second level that operates on the sequence of word representations to predict the language tag for each word. In the first level, the model uses convolutional neural network layers to do a soft-version of n-gram matching. The output of this layer is a feature vector that provides a useful signal for the language of each word because languages tend to differ in their character n-gram distributions. The second level of the model is a bidirectional LSTM that takes as input the feature vectors from the previous layer and outputs the predicted tag for each word. The use of the LSTM allows the model to incorporate evidence from arbitrary far away in the word sequence. We made one tweak that was not described in Jaech et al. (2016b): the standard LSTM was replaced with an LSTM that has coupled input and forget gates for a 25% reduction in the parameters in the bi-LSTM and a corresponding improvement in speed of computation (Greff et al., 2016). Operating on the word-level representations allows the LSTM to predict the correct tag for words whose language is ambiguous from just the character-level feature vectors based on the fact that adjacent words are more likely to belong to the same language.

We tune the model hyper-parameters by training and testing on the train and dev splits of the Workshop dataset, effectively making the task harder for the model trained on the Anchored corpus. Table 6 shows the word-level and sentence-level accuracy and the average F1-score of the language ID tagset for each training/testing combination.

First, we trained our tagger on the Workshop data (Workshop Model, in Table 6) and observed that its performance on the Workshop test set is similar to that reported for this model in the Shared Task (95.93%). The performance of this tagger however sees a big drop of performance on word-level accuracy and sentence-level accuracy when tested on the Anchored test set. This demonstrates that a tagger trained on a corpus comprised of majority of monolingual sentences, with a lower degree of bilingualism and switching rates, has some difficulty generalizing to a more balanced corpus like the Anchored Tweets Corpus.

Second, we partitioned the Anchored corpus in train and test by randomly choosing 1,500 tweets for the test set and leaving the rest for training. We trained a new tagger on the Anchored dataset with the same hyper-parameter settings as the Workshop tagger and report its test performance on Table 6 as Anchored tagger. We observed that the performance of this model on the Workshop data is very good, despite the difference between the two datasets: the word-level accuracy only decreases by 0.8% accuracy points with respect to the Workshop model, whereas the sentence-level

Training Corpus	Word Accuracy (%)		Avg. F1-Score		Sentence Accuracy (%)	
	Workshop	Anchored	Workshop	Anchored	Workshop	Anchored
Workshop	95.93	82.09	0.4218	0.3978	67.91	14.20
Anchored	95.13	**91.86**	**0.4655**	**0.5937**	62.60	**40.13**
Combination	**96.91**	91.61	0.4328	0.5617	**73.53**	39.87

Table 6: Language tagging accuracy (left) and average f1-score (center) at the word level and language tagging accuracy at the sentence-level (right) for each training and testing combination.

Training Corpus	Word Accuracy (%)		Avg. F1-Score		Fragment Accuracy (%)	
	Workshop	Anchored	Workshop	Anchored	Workshop	Anchored
Workshop	85.46	78.96	0.3678	0.3802	84.29	61.67
Anchored	83.85	86.64	0.3617	0.4937	82.61	71.73
Combination	87.44	86.98	0.3722	0.5020	86.51	73.67

Table 7: Language tagging accuracy (left) and average f1-score (center) at the word level and language tagging accuracy at the fragment-level (right) for each training and testing combination on the subset of code-switched fragments.

accuracy decreases by 5.31% points. However the F1-score value sees a relative improvement of 10.36%, which indicates that the new corpus is more similar to the Workshop test split than the Workshop train-dev split. The Anchored-trained tagger achieves 91.86% word-level accuracy on its own test set, with 0.5937 average F1-score value and 40.13% sentence-level accuracy. These results indicate that a tagger trained on the anchored corpus is able to generalize quite well on the same corpus, although overall the classification task is harder than on the Workshop corpus: the best word-level and sentence-level accuracies in the Workshop test set are much higher than in the Anchored test set.

Finally, we trained a tagger on a combination of the Workshop and Anchored training sets. This combined tagger achieves the best word-level accuracy on the Workshop corpus (96.91%) as shown in the last row of Table 6. Similarly the combined tagger also achieves the best sentence-level accuracy on the Workshop test set (73.53%).

Overall, the Anchored tagger achieves the best results on the Anchored test set for every metric (91.86% word-level accuracy, 0.5937 average f1-score and 40.13% sentence-level accuracy), despite being trained on much less data (the anchored train set has 7,025 tweets, the workshop train set has 11,400 tweets and the combined train set has 18,425 tweets). It also achieves the best average f1-score on the Workshop test set (0.4655). The Combination tagger achieves the best word-level and sentence-level accuracy on the Workshop test set (96.91% and 73.53% respectively).

We next examine the performance of the three taggers on the subset of code-switched segments present in each test set in Table 7, where we define a code-switched segment as the minimal span of tokens where a point code-switch occurs. Notice that a segment can be longer than two tokens if there is a Named Entity, Other, Mixed or Ambiguous token in between. For example, from the sentence "I watched The Godfather y me encantó", the code-switched segment would be "watched The Godfather y" where "The Godfather" is a Named Entity.

From this table we can see that, in fact, taggers have most difficulty tagging words that occur in the context of a code-switch, since the accuracy of all three models on both test subsets of code-switched segments suffers a steep decline for the results shown for the complete test set in the left subtable of Table 6. In the case of the Workshop tagger, its accuracy has relative changes of -10.91 and -3.81% on the full workshop and anchored test sets respectively. The Anchored model sees even larger relative decreases of -11.86 and -5.68%. In comparison, the Combination model has the smallest relative decreases in accuracy, with -9.77 and -5.05%. The same trends can be observed for the average F1-Score and the fragment-level accuracy metrics.

Overall the best performing model is the one trained on the combined training sets, followed by the Anchored model, which always gets better metric values on its own test set and achieves similar metric values on the Workshop test set when compared to the Workshop tagger. Notice though that the Anchored model was trained on less than 40% of the number of tweets in the Combined train set.

7. Conclusions

In this paper we present a method, which makes use of *anchoring* and monolingual Language ID, for detecting code-switched text. We relax strict anchoring constraints to query the Twitter API and retrieve code-switched tweets. We crowdsource language tags for the tokens of 8,285 tweets and found that almost 70% of the collected tweets are indeed code-switched. These tweet exhibit a relatively balanced amount of Spanish and English text and a high amount of code-switching per tweet. The average number of code-switches per tweet in the corpus is 1.19 switches while 7.77% of the tokens are followed by a code-switch. These numbers compare favorably to the 2016 EMNLP Workshop Shared Task Code-Switched Twitter corpus, which was obtained with a different, more labor-intensive method. We evaluated the quality of our new Anchored corpus by training state-of-the-art language taggers and showed that a) a tagger trained on the original Workshop corpus exhibits a more considerable drop in accuracy when tested on the Anchored corpus; and b) a tagger trained on the Anchored corpus achieves very good accuracy on both test corpora. These results show great promise for automatic collection of other code-switched corpora for use in training language models and for other NLP and speech tasks.

8. Bibliographical References

Adel, H., Vu, N. T., Kraus, F., Schlippe, T., Li, H., and Schultz, T. (2013a). Recurrent neural network language modeling for code switching conversational speech. In *Proc. of ICASSP*, pages 8411–8415. IEEE.

Adel, H., Vu, N. T., and Schultz, T. (2013b). Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proc. of ACL*, pages 206–211.

Ahmed, B. H. and Tan, T.-P. (2012). Automatic speech recognition of code switching speech using 1-best rescoring. In *Proc. of IALP*, pages 137–140.

Al-Badrashiny, M. and Diab, M. (2016). The George Washington University system for the code-switching workshop shared task 2016. In *Proc. of the Second Workshop on Computational Approaches to Code Switching*, pages 108–111.

AlGhamdi, F., Molina, G., Diab, M., Solorio, T., Hawwari, A., Soto, V., and Hirschberg, J. (2016). Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107.

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

Çetinoglu, Ö. (2016). A Turkish-German codeswitching corpus. In *Proc. of LREC*, pages 4215–4220.

Elfardy, H., Al-Badrashiny, M., and Diab, M. (2014). AIDA: identifying code switching in informal arabic text. In *Proc. of The First Workshop on Computational Approaches to Code Switching*, pages 94–101.

Franco, J. C. and Solorio, T. (2007). Baby-steps towards building a Spanglish language model. In *Proc. of International Conference on Intelligent Text Processing and Computational Linguistics*, pages 75–84. Springer.

Go, A., Huang, L., and Bhayani, R. (2009). Twitter sentiment analysis. *Entropy*, 17:252.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proc. of LREC*, pages 759–765.

Goyal, P., Mital, M. R., and Mukerjee, A. (2003). A bilingual parser for Hindi, English and code-switching structures. In *EACL Workshop of Computational Linguistics for South Asian Languages*.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*.

Jaech, A., Mulcaire, G., Hathi, S., Ostendorf, M., and Smith, N. A. (2016a). Hierarchical character-word models for language identification. *arXiv preprint arXiv:1608.03030*.

Jaech, A., Mulcaire, G., Hathi, S., Ostendorf, M., and Smith, N. A. (2016b). A neural model for language identification in code-switched tweets. In *Proc. of the Second Workshop on Computational Approaches to Code Switching*, pages 60–64.

Jamatia, A., Gambäck, B., and Das, A. (2015). Part-of-speech tagging for code-mixed English-Hindi Twitter and Facebook chat messages. In *Proc. of Recent Advances in Natural Language Processing*, pages 239–248.

Lee, S. Y. M. and Wang, Z. (2015). Emotion in code-switching texts: Corpus construction and analysis. *Proc. of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 91–99.

Li, Y. and Fung, P. (2012). Code-switch language model with inversion constraints for mixed language speech recognition. In *Proc. of COLING 2012*, pages 1671–1680.

Li, Y. and Fung, P. (2014). Language modeling with functional head constraint for code switching speech recognition. In *Proc. of EMNLP*, pages 907–916.

Lui, M. and Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *Proc. of the ACL*, pages 25–30.

Lyudovyk, T. and Pylypenko, V. (2014). Code-switching speech recognition for closely related languages. In *Proc. of SLTU*, pages 188–193.

Mendels, G., Cooper, E., Soto, V., Hirschberg, J., Gales, M. J., Knill, K. M., Ragni, A., and Wang, H. (2015). Improving speech recognition and keyword search for low resource languages using web data. In *Proc. of INTERSPEECH*, pages 829–833.

Mendels, G., Cooper, E., and Hirschberg, J. (2016). Babler-data collection from the web to support speech recognition and keyword search. In *Proc. of Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 72–81.

Molina, G., Rey-Villamizar, N., Solorio, T., AlGhamdi, F., Ghoneim, M., Hawwari, A., and Diab, M. (2016). Overview for the second shared task on language identification in code-switched data. In *Proc. of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Piergallini, M., Shirvani, R., Gautam, G. S., and Chouikha, M. (2016). The Howard University system submission for the shared task in language identification in spanish-english codeswitching. In *Proc. of the Second Workshop on Computational Approaches to Code Switching*, pages 116–120.

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., and Solorio, T. (2016). Multilingual code-switching identification via LSTM recurrent neural networks. In *Proc. of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, TX.

Samih, Y. (2016). An Arabic-Moroccan darija code-switched corpus. In *Proc. of LREC*, pages 4170–4175.

Solorio, T. and Liu, Y. (2008a). Learning to predict code-switching points. In *Proc. of EMNLP*, pages 973–981.

Solorio, T. and Liu, Y. (2008b). Part-of-speech tagging for English-Spanish code-switched text. In *Proc. of EMNLP*, pages 1051–1060.

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Gohneim, M., Hawwari, A., AlGhamdi, F., Hirschberg,

J., Chang, A., et al. (2014). Overview for the first shared task on language identification in code-switched data. In *Proc. of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Tiedemann, J. and Ljubešić, N. (2012). Efficient discrimination between closely related languages. In *COLING 2012*, pages 2619–2634.

Vilares, D., Alonso, M. A., and Gómez-Rodrıguez, C. (2015). Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proc. of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, pages 2–8.

Vyas, Y., Gella, S., and Sharma, J. (2014). POS tagging of English-Hindi code-mixed social media content. In *Proc. of EMNLP*, pages 974–979.

9. Language Resources References

Consortium, L. D. et al. (2003). English Gigaword. `http://www.ldc.upenn.edu`.

Graff, D. (2011). Spanish Gigaword third edition (ldc2011t12).