The Tel Aviv University System for the Code-Switching Workshop Shared Task

Kfir Bar
School of Computer Science
Tel Aviv University
Ramat Aviv, Israel
kfirbar@post.tau.ac.i

Nachum Dershowitz
School of Computer Science
Tel Aviv University
Ramat Aviv, Israel
nachumd@tau.ac.il

Abstract

We describe our entry in the EMNLP 2014 code-switching shared task. Our system is based on a sequential classifier, trained on the shared training set using various character- and word-level features, some calculated using a large monolingual corpora. We participated in the Twitter-genre Spanish-English track, obtaining an accuracy of 0.868 when measured on the tweet level and 0.858 on the word level.

1 Introduction

Code switching is the act of changing language while speaking or writing, as often done by bilinguals (Winford, 2003). Identifying the transition points is a necessary first step before applying other linguistic algorithms, which usually target a single language. A switching point may occur between sentences, phrases, words, or even between certain morphological components. Code switching happens frequently in informal ways of communication, such as verbal conversations, blogs and microblogs; however, there are many examples in which languages are switched in formal settings. For example, alternating between Colloquial Egyptian Arabic and Modern Standard Arabic in modern Egyptian prose is prevalent (Rosenbaum, 2000).

This shared task (Solorio et al., 2014),[1] the first of its kind, challenges participants with identifying those switching points in blogs as well as in microblog posts. Given posts with a mix of a specific pair of languages, each participating system is required to identify the language of every word. Four language-pair tracks were offered by the task organizers: Spanish-English, Nepali-English, Modern Standard Arabic and Colloquial Arabic, and Mandarin-English. For each language pair, a training set of Twitter[2] statuses was provided, which was manually annotated with a label for every word, indicating its language. In addition to the two language labels, a few additional labels were used. Altogether there were six labels: (1) lang1—the first language; (2) lang2—the second language; (3) ne—named entity; (4) ambiguous—for ambiguous words belonging to both languages; (5) mixed—for words composed of morphemes in each language; and (6) other—for cases where it is impossible to determine the language. For most of the language pairs, the organizers supplied three different evaluation sets. The first set was composed of a set of unseen Twitter statuses, provided with no manual annotation. The other two sets contained data from a "surprise genre", mainly composed of blog posts.

We took part only in the Spanish-English track. Both English and Spanish are written in Latin script. The Spanish alphabet contains some additional letters, such as those indicating stress (vowels with acute accents: á, é, í, ó, ú), a u adorned with a diaeresis (ü), the additional letter ñ (*eñe*), and inverted question and exclamation punctuation marks ¿ and ¡ (used at the beginning of questions and exclamatory phrases, respectively). Although social-media users are not generally consistent in their use of accents, their appearance in a word may disclose its language. By and large, algorithms for code switching have used the character-based k-mer feature, introduced by (Cavnar and Trenkle, 1994).[3]

Our system is an implementation of a multiclass classifier that works on the word level, considering features that we calculate using large Spanish as well as English monolingual corpora. Working with a sequential classifier, the predicted

[1] http://emnlp2014.org/workshops/CodeSwitch/call.html

[2] http://www.twitter.com

[3] We propose the term "k-mer" for character k-grams, in contradistinction to word n-grams.

Proceedings of The First Workshop on Computational Approaches to Code Switching, pages 139–143,
October 25, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

labels of the previous words are used as features in predicting the current word.

In Section 2, we describe our system and the features we use for classification. Section 3 contains the evaluation results, as published by the organizers of this shared task. We conclude with a brief discussion.

2 System Description

We use a supervised framework to train a classifier that predicts the label of every word in the order written. The words were originally tokenized by the organizers, preserving punctuation, emoticons, user mentions (e.g., @emnlp2014), and hashtags (e.g., #emnlp2014) as individual tokens. The informal language, as used in social media, introduces an additional challenge in predicting the language of every word. Spelling mistakes as well as grammatical errors are very common. Hence, we believe that predicting the language of a given word merely using dictionaries for the two languages is likely to be insufficient.

Our classifier is trained on a learning set, as provided by the organizers, enriched with some additional features. Every word in the order written is treated as a single instance for the classifier, each including features from a limited window of preceding and successive words, enriched with the predicted label of some of the preceding words. We ran a few experiments with different window sizes, based on 10-fold cross validation, and found that the best token-level accuracy is obtained using a window of size 2 for all features, that is, two words before the focus word and two words after.

The features that we use may be grouped in three main categories, as described next.

2.1 Features

We use three main groups of features:

Word level: The specific word in focus, as well as the two previous words and the two following ones are considered as features. To reduce the sparsity, we convert words into lowercase. In addition, we use a monolingual lexicon for English words that are typically used in Twitter. For this purpose, we employ a sample of the Twitter General English lexicon, released by Illocution, Inc.,[4] containing the top 10K words and bigrams from a relatively large corpus of public English tweets they collected over a period of time, along with frequency information. We bin the frequency rates into 5 integer values (with an additional value for words that do not exist in the lexicon), which are used as the feature value for every word in focus, and for the other four words in its window. This feature seems to be quite noisy, as some common Spanish words appear in the lexicon (e.g., *de*, *no*, *a*, *me*); on the other hand, it may capture typical English misspellings and acronyms (e.g., *oomf*, *noww*, *lmao*). We could not find a similar resource for Spanish, unfortunately.

To help identify named entities, we created a list of English as well Spanish names of various entity types (e.g., locations, family and given names) and used it to generate an additional boolean feature, indicating whether the word in focus is an entity name. The list was compiled out of all words beginning with a capital letter in relatively large monolingual corpora, one for English and another for Spanish. To avoid words that were capitalized because they occur at the beginning of a sentence, regardless of whether they are proper names, we first processed the text with a true-casing tool, provided as part of Moses (Koehn et al., 2007)—the open source implementation for phrase-based statistical machine translation. Our list contains about 146K entries.

Intra-word level: Spanish, as opposed to English, is a morphologically rich language, demonstrating a complicated suffix-based derivational morphology. Therefore, in order to capture repeating suffixes and prefixes that may characterize the languages, we consider as features substrings of 1–3 prefix and suffix characters of the word in focus and the other four words in its window. Although it is presumed that capitalization is not used consistently in social media, we consider a boolean feature indicating whether the first letter of each word in the window was capitalized in the original text or not. At this level, we use two additional features that capture the level of uncertainty of seeing the sequence of characters that form the specific word in each language. This is done by employing a 3-mer character-based language model, trained over a large corpus in each language. Then, the two language models, one for each language, are applied on the word in focus to calculate two log-probability values. These are binned into ten discrete values that are used as the features' values. We add a boolean feature, indi-

[4] http://www.illocutioninc.com

cating which of the two models returned a lower log probability.

Inter-word level: We capture the level of uncertainty of seeing specific sequences of words in each language. We used 3-gram word-level language models, trained over large corpora in each of the languages. We apply the models to the focus word, considering it to be the last in a sequence of three words (with the two previous words) and calculate log probabilities. Like before, we bin the values into ten discrete values, which are then used as the features' values. An additional boolean feature is used, indicating which of the two models returned a lower log probability.

2.2 Supervised Framework

We designed a sequential classifier running on top of the Weka platform (Frank et al., 2010) that is capable of processing instances sequentially, similar to YamCha (Kudo and Matsumoto, 2003). We use LibSVM (Chang and Lin, 2011), an implementation of Support Vector Machines (SVM) (Cortes and Vapnik, 1995), as the underlying technology, with a degree 2 polynomial kernel. Since we work on a multi-class classification problem, we take the one-versus-one approach. As mentioned above, we use features from a window of ±2 words before and after the word of interest. In addition, for every word, we consider as features the predicted labels of the two prior words.

3 Evaluation Results

We report on the results obtained on the unseen task evaluation sets, which were provided by the workshop organizers.[5] There are three evaluation sets. The first is composed of a set of unseen Twitter statuses and the other two contain data from a "surprise genre". The results are available online at the time of writing only for the first and second sets. The results of the third set will be published during the upcoming workshop meeting.

The training set contains 11,400 statuses, comprising 140,706 words. Table 1 shows the distribution of labels.

The first evaluation set contains 3,060 tweets. However, we were asked to download the statuses directly from Twitter, and some of the statuses were missing. Therefore, we ended up with only 1,661 available statuses, corresponding to 17,723 words. According to the organizers, the evaluation was performed only on the 1,626 tweets that were available for all the participating groups. Out of the 1,626, there are 1,155 monolingual tweets and 471 code-switched tweets. Table 2 shows the evaluation results for the Tel Aviv University (TAU) system on the first set, reported on the tweet level.

Label	**Number**
lang1	77,101
lang2	33,099
ne	2,918
ambiguous	344
mixed	51
other	27,194

Table 1: Label distribution in the training set.

Accuracy	0.868
Recall	0.720
Precision	0.803
F1-Score	0.759

Table 2: Results for the first evaluation set, measured on tweet level.

In addition, the organizers provide evaluation results, calculated on the word level. Table 3 shows the label distribution among the words in the first evaluation set, and Table 4 shows the actual results. The overall accuracy on the word level is 0.858.

The second evaluation set contains 1,103 words of a "surprise" (unseen) genre, mainly blog posts. Out of the 49 posts, 27 are monolingual and 22 are code-switched posts. Table 5 shows the results for the surprise set, calculated on the post level.

As for the first set, Table 6 shows the distribution of the labels among the words in the surprise set, and in Table 7 we present the results as measured on the word level. The overall accuracy on the surprise set is 0.941.

4 Discussion

We believe that we have demonstrated the potential of using sequential classification for code-switching, enriched with three types of features, some calculated using large monolingual corpora. Compared to the other participating systems as published by the workshop organizers, our system obtained encouraging results. In particular, we observe relatively good results in relating words to

[5] `http://emnlp2014.org/workshops/CodeSwitch/results.php`

Label	Count
lang1 (English)	7,040
lang2 (Spanish)	5,549
ne	464
mixed	12
ambiguous	43
other	4,311

Table 3: Label distribution in the first evaluation set.

Label	Recall	Precision	F1-Score
lang1 (English)	0.900	0.830	0.864
lang2 (Spanish)	0.869	0.914	0.891
ne	0.313	0.541	0.396
mixed	0.000	1.000	0.000
ambiguous	0.023	0.200	0.042
other	0.845	0.860	0.853

Table 4: Results for the first evaluation set, measured on word level.

their language; however, identifying named entities did not work as well. We plan to further investigate this issue. The results on the surprise genre are similar to that for the genre the system was trained on. However, since the surprise set is relatively small in size, we refrain from drawing conclusions about this. Trying the same code-switching techniques on other pairs of languages is part of our planned future research.

Accuracy	0.864
Recall	0.708
Precision	0.803
F1-Score	0.753

Table 5: Results for the second, "surprise" evaluation set, measured on the post level.

Label	Count
lang1 (English)	636
lang2 (Spanish)	306
ne	38
mixed	1
ambiguous	1
other	120

Table 6: Label distribution in the "surprise" evaluation set.

Label	Recall	Precision	F1-Score
lang1 (English)	0.883	0.824	0.853
lang2 (Spanish)	0.864	0.887	0.876
ne	0.293	0.537	0.379
mixed	0.000	1.000	0.000
ambiguous	0.022	0.200	0.039
other	0.824	0.843	0.833

Table 7: Results for the "surprise" evaluation set, measured on the word level.

References

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)*, pages 161–175.

Chih C. Chang and Chih J. Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, May.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20:273–297.

Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, Ian H. Witten, and Len Trigg. 2010. Weka—A machine learning workbench for data mining. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, chapter 66, pages 1269–1277. Springer US, Boston, MA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Interactive Poster and Demonstration Sessions of the 45th Annual Meeting of the ACL (ACL '07)*, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taku Kudo and Yuji Matsumoto. 2003. Fast methods for kernel-based text analysis. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 24–31, Sapporo, Japan.

Gabriel M. Rosenbaum. 2000. Fushammiyya: Alternating style in Egyptian prose. *Journal of Arabic Linguistics (ZAL)*, 38:68–87.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steve Bethard, Mona Diab, Mahmoud Gonheim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirshberg, 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. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar.

Donald Winford, 2003. *Code Switching: Linguistic Aspects*, chapter 5, pages 126–167. Blackwell Publishing, Malden, MA.