Language Identification in Code-Switched Text Using Conditional Random Fields and Babelnet

Utpal Kumar Sikdar and **Björn Gambäck**
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
{sikdar.utpal,gamback}@idi.ntnu.no

Abstract

The paper outlines a supervised approach to language identification in code-switched data, framing this as a sequence labeling task where the label of each token is identified using a classifier based on Conditional Random Fields and trained on a range of different features, extracted both from the training data and by using information from Babelnet and Babelfy.

The method was tested on the development dataset provided by organizers of the shared task on language identification in code-switched data, obtaining tweet level monolingual, code-switched and weighted F1-scores of 94%, 85% and 91%, respectively, with a token level accuracy of 95.8%. When evaluated on the unseen test data, the system achieved 90%, 85% and 87.4% monolingual, code-switched and weighted tweet level F1-scores, and a token level accuracy of 95.7%.

1 Introduction

Today many short messages contain words from different languages and it is a challenging task to identify which languages the different words are written in. Often the messages contain text snippets from several languages, that is, showing code-switching. Sometimes the messages even contain code-mixing, where there is a mix of the languages inside a single utterance or even inside a token itself.

The first code-switching data challenge was organized at EMNLP 2014 (Solorio et al., 2014). The task was to identify the language for each word in a text, classifying the words according to six labels: 'Lang1', 'Lang2', 'Mixed', 'NE', 'Other', and 'Ambiguous'. The first two labels identify tokens from the main languages that are mixed in the text, while the third is for tokens with word-internal mixing between these languages; 'NE' for named entities; 'Other' for language independent tokens (punctuation, numbers, etc.) and tokens from other languages, and 'Ambiguous' denotes tokens that cannot safely be assigned any (or only one) of the other labels. This shared task was organized again this year (Molina et al., 2016), with new datasets and slightly different labels, adding 'Unk' for unknown tokens.[1]

Work on developing tools for automatic language identification was initiated already in the 1960s (Gold, 1967), and although analysing code-switched text is a research area which has started to achieve wide-spread attention only in recent years, the first work in the field was carried out over thirty years ago by Joshi (1982), while Bentahila and Davies (1983) examined the syntax of the intra-sentential code-switching between Arabic and French. They claimed that Arabic-French code-switching was possible at all syntactic boundaries above the word level.

Das and Gambäck (2013) give a comprehensive overview of the work on code-switching until 2015. Notably, Solorio and Liu (2008) trained classifiers to predict code-switching points in Spanish and English, using different learning algorithms and transcriptions of code-switched discourse, while Nguyen and Doğruöz (2013) focused on word-level language identification (in Dutch-Turkish news commentary). Nguyen and Cornips (2016) describe

[1] An eighth label 'FW' was included for foreign words, but no words in the English-Spanish corpora were tagged with it.

Proceedings of the Second Workshop on Computational Approaches to Code Switching, pages 127–131,
Austin, TX, November 1, 2016. ©2016 Association for Computational Linguistics

work on analyzing and detecting intra-word code-mixing by first segmenting words into smaller units and later identifying words composed of sequences of subunits associated with different languages in *tweets* (posts on the Twitter social-media site).

The paper is organized as follows: Section 2 provides a description of the language identification method, whereby a supervised model was built using Conditional Random Fields to classify each token in a tweet into one of the seven categories based on different features, most of which are extracted from the training data, as described in Section 3. Results are then presented and discussed in Section 4, while Section 5 addresses future work and concludes.

2 Language Identification Method

The language identification system was built around a Conditional Random Field (CRF) classifier. We used the C^{++}-based CRF^{++} package (Kudo, 2013), a simple, customizable, and open source implementation of Conditional Random Fields for segmenting or labelling sequential data. Conditional Random Fields (Lafferty et al., 2001) are conditional, undirected graphical models that can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training.

Conditional Random Fields calculate the conditional probability of values on designated output nodes given the values of (other) designated input nodes. The conditional probability of a state sequence $s =< s_1, s_2, \ldots, s_T >$ given an observation sequence $o =< o_1, o_2, \ldots, o_T >$ is calculated as in Equation 1 (McCallum, 2003):

$$P_\wedge(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)) \tag{1}$$

where $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k, is to be learned via training. The feature function values may range from $-\infty$ to $+\infty$.

To make all the conditional probabilities sum up to 1, the normalization factor Z_o is calculated in the same fashion as in HMMs (Hidden Markov Models), that is, as given by Equation 2.

$$Z_o = \sum_s \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)) \tag{2}$$

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L_\wedge = \sum_{i=1}^{N} \log(P_\wedge(s^{(i)}|o^{(i)})) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \tag{3}$$

where $\{< o^{(i)}, s^{(i)} >\}$ is the labelled training data. The second sum corresponds to a zero-mean, σ^2-variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set the parameter λ to maximize the penalized log-likelihood using Limited-memory BFGS (Sha and Pereira, 2003), a quasi-Newton method that is highly efficient, and which results in only minor changes in accuracy due to changes in λ.

2.1 Features based on training data

Two sets of features were developed to train the model: one extracted from the training data and the other based on information from Babelnet (Navigli and Ponzetto, 2012) and Babelfy (Moro et al., 2014), with most of the features and their settings being based on the training data. The complete set of features induced from training data was as follows:

Local context. Local contexts play an important role for identifying the languages. Here the two preceding and two succeeding words were used as local context.

Word suffix and prefix. Fixed length characters stripped from the beginning and ending of the current word. Up to 4 characters were removed.

Word length. Analysis of the training data showed that the Spanish words on average were shorter than the English words. Words with 1–4 characters were flagged with a binary feature.

Word previously occurred. A binary feature which checks if a word already occurred in the training data or not.

Initial capital. In general, proper nouns tend to start with capital letters, so this feature checks whether the current word has an initial capital.

All capitals. A binary feature which is set if the current word contains only capital letters. The feature is very helpful for identifying named entities (since, e.g., abbreviations often refer to named entities).

Single capital letter: checks if the word contains a single capital letter or not.

All digits: set to 1 if the word contains only numerical characters. This is helpful for identifying tokens belonging to the 'Other' category.

Alphanumeric: a binary feature which flags if the word contains only digits and alphabetical characters together.

All English alphabet: checks if all a word's characters belong to the English alphabet.

Special Spanish character: a flag which is set if the current word contains any Spanish-specific letters (á, é, etc.).

Hash symbol: set to 1 if a word contains the symbol '#', otherwise 0.

Rate symbol: set to 1 if the current word contains the symbol '@'.

Word with single letter. Many single letter words were observed to belong to Spanish, so this flag is set if the word length is exactly 1.

Two consecutive letters. Some words repeat two character sequences several times (e.g., *hahaha*, *jaja*). Each token is split into two character sequences and this binary feature is set if each two letter character sequences matches.

Same letter occurred multiple times. Many words in tweets contain sequences of the same character repeated many times (e.g., ewww, yaaaas). The feature is set if a letter occurred in a word more than two times consecutively.

Gazetteer NE list. A list of named entities (NE) was collected from the training data. This flag is set if a token matches an item on the NE list.

Special character list. A list of special characters (e.g., emojis) was collected from the training data. If a tokens contains any character which is on the list, the binary feature is set.

2.2 Babelnet and Babelfy features

Three further features were developed from external resource, Babelnet (Navigli and Ponzetto, 2012) and Babelfy (Moro et al., 2014):

Dataset	Number of	
	tweets	tokens
Training	11,400	140,745
Development	3,014	33,743
Test	18,237	218,138

Table 1: Statistics of the tweet datasets

WordNet feature: Every token is passed to the Babelnet database for checking whether the token exists in the English WordNet or not. If the token appears in the database, the feature is set to 1, otherwise to 0.

Multilingual WordNet: The Babelnet Multilingual WordNet is checked for Spanish, by passing each token to the Babelnet database and checking whether the token is present in the database or not.

Babelfy Named Entity: Named entities are extracted from Babelfy and used as a feature, which is utilized for identification of the 'NE' category tokens.

3 Datasets

We used the datasets provided by the organizers of the EMNLP 2016 code-switching workshop shared task on language identification in code-switched data (Molina et al., 2016).[2]

Three types of data were provided: training, development and test. In the training and development datasets, the total number of tweets are 11,400 and 3,014, respectively, with language identification offsets given for each category. In the test data, the total number of tweets is 18,237 without annotations.

The number of tweets and the number of tokens in each of the three datasets are given in Table 1.

4 Results

We developed a supervised model for language identification using the CRF++ classifier, implemented the different features described above, and trained the CRF++ classifier using these features. Initially, the classifier was trained using the training data and tested on the development data for Spanish-English.

[2] `care4lang1.seas.gwu.edu/cs2/call.html`

System setup	Mono-lingual			Code-switched			Weighted	Token-level
	P	R	F_1	P	R	F_1	F_1	Accuracy
Without external resources	0.94	0.92	0.93	0.82	0.85	0.83	0.904	0.943
With external resources	0.95	0.93	0.94	0.82	0.87	0.85	0.911	0.952

Table 2: System performance on the development data, with and without the external resources (Babelnet and Babelfy)

Team	Accuracy
IIIT Hyderabad	0.961
NepSwitch	0.958
NTNU	0.957
HHU-UH-G	0.953
Howard U	0.951
McGill U	0.941
UW group	0.926
GWU*	0.918
Arunavha Chana	0.527

Team	Mono-lingual	Code-switched	Weighted
Howard U	0.90	0.87	0.890
IIIT Hyderabad	0.90	0.86	0.886
HHU-UH-G	0.90	0.85	0.878
NTNU	0.90	0.85	0.874
NepSwitch	0.89	0.85	0.870
McGill U	0.86	0.78	0.820
UW group	0.78	0.78	0.780
Arunavha Chana	0.80	0.71	0.760
GWU*	0.93	0.54	0.740

Table 3: Token level accuracy (left) and tweet level F_1 scores (right) on the test data for all participating systems

There were two types of evaluation, at tweet level and at token level. The tweet level precision (P), recall (R) and F_1-scores obtained on the monolingual part of the development data were 95%, 93% and 94%, respectively. On the code-switched part of that data, the precision, recall and F_1-scores were 82%, 87% and 85%, giving a weighted, total F_1-score of 91.1%. For token level evaluation, the development data accuracy was 95.2%, as shown in Table 2.

Table 2 also gives the development data scores for a system trained without the second feature set, i.e., without the Babelnet and Babelfy features. As can be seen in the table, the contribution from those features is small but useful, adding 0.9% to the token-level accuracy and 0.7% to the tweet-level weighted F_1 score, with the main contribution (2%) being on recall for the tweets containing code-switching.

Applying our system (NTNU) to the test data, the tweet level monolingual, code-switched and weighted F_1-scores were 90%, 85% and 87.4%, with a token level accuracy performance of 95.7%.

A comparison of the results of the different systems participating in the shared task is given in Table 3, for both token level and tweet level evaluation, with the performance of our system marked in bold face. For token level evaluation, the NTNU system achieved third place in the shared task, with an accuracy difference between our system and the best performing system (IIIT Hyderabad) of only 0.4%.

At the tweet level, the NTNU system performed on par with the best systems on the monolingual tweets, while it scored 2% lower on the tweets that contained some code-switching, giving it fourth place on weighted F_1-score. However, as can be seen from the tables, the top-5 teams actually obtained very similar performance on all measures, and both at token and tweet level.

5 Conclusion

For this shared task, we have outlined an approach using a CRF based system for language identification in code-switched data, implementing a range of different features and achieving state-of-the-art results. Most of the features are extracted directly from training data, while some are induced by using Babelnet and Babelfy as external resources.

In future, we will aim to optimize these features using grid search and evolutionary algorithms, as well as generate different models using several classification algorithms and utilize the predictions of ensembles of such machine learners in order to enhance the overall system performance.

References

Abdelâli Bentahila and Eirlys E. Davies. 1983. The syntax of Arabic-French code-switching. *Lingua*, 59:301–330.

Amitava Das and Björn Gambäck. 2013. Code-mixing in social media text: The last language identification frontier? *Traitement Automatique des Langues*, 54(3):41–64.

E. Mark Gold. 1967. Language identification in the limit. *Information and Control*, 10(5):447–474.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th International Conference on Computational Linguistics*, pages 145–150, Prague, Czechoslovakia, July. ACL.

Taku Kudo. 2013. CRF++: Yet another CRF toolkit. `http://taku910.github.io`.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Maryland, USA, June.

Andrew McCallum. 2003. Efficiently inducing features of Conditional Random Fields. In *Proceedings of the 19th Conference on Uncertainty in Articifical Intelligence*, pages 403–410, Acapulco, Mexico, August. Association for Uncertainty in Artificial Intelligence.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, November. ACL. 2nd Workshop on Computational Approaches to Linguistic Code Switching.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Dong Nguyen and Leonie Cornips. 2016. Automatic detection of intra-word code-switching. In *Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 82–86.

Dong Nguyen and A. Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 857–862, Seattle, Washington, October. ACL.

Fei Sha and Fernando C. N. Pereira. 2003. Shallow parsing with Conditional Random Fields. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 134–141, Edmonton, Alberta, May. ACL.

Thamar Solorio and Yang Liu. 2008. Learning to predict code-switching points. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 973–981, Honolulu, Hawaii, October. ACL.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Doha, Qatar, October. ACL. 1st Workshop on Computational Approaches to Code Switching.