LILI: A Simple Language Independent Approach for Language Identification

Mohamed Al-Badrashiny and Mona Diab
Department of Computer Science, The George Washington University
{badrashiny,mtdiab}@gwu.edu

Abstract

We introduce a generic Language Independent Framework for Linguistic Code Switch Point Detection. The system uses the word length, character level (1, 2, 3, 4, and 5)-grams and word level unigram language models to train a conditional random fields (CRF) model for classifying input words into various languages. We test our proposed framework and compare it to the state-of-the-art published systems on standard data sets from several language pairs: English-Spanish, Nepali-English, English-Hindi, Arabizi (Refers to Arabic written using the Latin/Roman script)-English, Arabic-Engari (Refers to English written using Arabic script), Modern Standard Arabic(MSA)-Egyptian, Levantine-MSA, Gulf-MSA, one more English-Spanish, and one more MSA-EGY. The overall weighted average F-score of each language pair are 96.4%, 97.3%, 98.0%, 97.0%, 98.9%, 86.3%, 88.2%, 90.6%, 95.2%, and 85.0% respectively. The results show that our approach despite its simplicity, either outperforms or performs at comparable levels to state-of-the-art published systems.

1 Introduction

Linguistic Code Switching (LCS) is a common practice among multilingual speakers in which they switch between their common languages in written and spoken communication. In Spanish-English for example: "She told me that mi esposo looks like un buen hombre." ("She told me that my husband looks like a good man"). In this work we care about detecting LCS points as they occur intra-sententially where words from more than one language are mixed in the same utterance. LCS is observed on all levels of linguistic representation, and especially pervasive in social media. LCS poses a significant challenge to NLP, hence detecting LCS points is a very important task for many downstream applications. In this paper we address this challenge using a generic simple language independent approach. We illustrate our approach on several language pairs utilizing publicly available data sets and comparing our performance against state-of-the-art sophisticated systems tailored to the problem of LCS point detection (LCSPD). Furthermore, we show the robustness of our approach on the most challenging problem of language variety code switching where the code switching is happening between a standard and dialect, namely we illustrate our performance on Modern Standard Arabic (MSA) mixed with Egyptian Dialectal Arabic data (EGY).

2 Related Work

Several systems have recently addressed the problem of LCSPD in written text both within language varieties and across different language pairs. Relevant work on the problem of LCSPD among different language pairs can be summarized in the following works.

3ARRIB (Al-Badrashiny et al., 2014; Eskander et al., 2014) addresses the challenge of how to distinguish between Arabic words written using Roman script (Arabizi) and actual English words in the same context/utterance. The assumption in this framework is that the script is Latin for all words. It trains a finite state transducer (FST) to learn the mapping between the Roman form of the Arabizi words and

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1211–1219, Osaka, Japan, December 11-17 2016.

their Arabic form. It uses the resulting FST to find all possible Arabic candidates for each word in the input text. These candidates are filtered using MADAMIRA (Pasha et al., 2014), a state-of-the-art morphological analyzer and POS disambiguation tool, to filter out non-Arabic solutions. Finally, it leverages a decision tree that is trained on language model probabilities of both the Arabic and Romanized forms to render the final decision for each word in context as either being Arabic or English.

Bar and Dershowitz (2014) addresses the challenge for Spanish-English LCSPD. The authors use several features to train a sequential Support Vector Machines (SVM) classifier. The used features include previous and following two words, substrings of 1-3 character ngrams from the beginning and end of each word thereby modeling prefix and suffix information, a boolean feature indicating whether the first letter is capitalized or not, and 3-gram character and word ngram language models trained over large corpora of English and Spanish, respectively.

Barman et al. (2014) present systems for both Nepali-English and Spanish-English LCSPD. The script for both language pairs is Latin based, i.e. Nepali-English is written in Latin script, and Spanish-English is written in Latin script. The authors carry out several experiments using different approaches including dictionary-based methods, linear kernel SVMs, and a k-nearest neighbor approach. The best setup they found is the SVM-based one that uses character n-gram, binary features indicate whether the word is in a language specific dictionary of the most frequent 5000 words they have constructed, length of the word, previous and next words, 3 boolean features for capitalization to check if the first letter is capitalized, if any letter is capitalized, or if all the letters are capitalized.

The approach presented by King et al. (2014) utilizes character n-gram probabilities, lexical probabilities, word label transition probabilities and existing named entity recognition tools within a Markov model framework.

Jain and Bhat (2014) use a CRF based token level language identification system that uses a set of easily computable features (Ex. isNum, isPunc, etc.). Their analysis showed that the most important features are the word n-gram posterior probabilities and word morphology.

Lin et al. (2014) use a CRF model that relies on character n-grams probabilities (tri and quad grams), prefixes, suffixes, unicode page of the first character, capitalization case, alphanumeric case, and tweet-level language ID predictions from two off-the-shelf language identifiers: cld2[1] and ldig.[2] They increase the size of the training data using a semi supervised CRF autoencoder approach (Ammar et al., 2014) coupled with unsupervised word embeddings.

MSR-India (Chittaranjan et al., 2014) uses character n-grams to train a maximum entropy classifier that identifies whether a word is language1 or language2. The resultant labels are then used together with word length, existence of special characters in the word, current, previous and next words to train a CRF model that predicts the token level classes of words in a given sentence/tweet.

On the other hand, for within language varieties, AIDA (Elfardy et al., 2014) and AIDA2 (Al-Badrashiny et al., 2015) are the best published systems attacking this problem in Arabic. In this context, the problem of LCSPD is more complicated than mixing two very different languages since in the case of varieties of the same language, the two varieties typically share a common space of cognates and often faux amis, where there are homographs but the words have very different semantic meanings, hence adding another layer of complexity to the problem. In this set up the assumed script is Arabic script. AIDA (Elfardy et al., 2014) uses a weakly supervised rule based approach that relies on a language model to tag each word in the given sentence. Then it uses the LM decision for each word in the given sentence/tweet and combine it with other morphological information to decide upon the final class of each word. AIDA2 (Al-Badrashiny et al., 2015) uses a complex system that is based on a mix of language dependent and machine learning components to detect the linguistic code switch between the modern standard Arabic (MSA) and Egyptian dialect (EGY) that are both written using Arabic script. It uses MADAMIRA (Pasha et al., 2014) to find the POS tag, prefix, lemma, suffix, for each word in the input text. Then it models these features together with other features including word level language model probabilities in a series of classifiers where it combines them in a classifier ensemble approach to

[1] https://code.google.com/p/cld2/

[2] https://github.com/shuyo/ldig

find the best tag for each word.

We compare our system to all of the above systems in addition to some other baselines.

3 Approach

In this paper, we present a very simple language independent framework called LILI to address the challenge of linguistic code switch point detection (LCSPD) when it occurs using the same script for the mixed languages in the utterance. Our framework is mainly based on the assumption that each language has its own character pattern behaviors and combinations relating to the underlying phonology, phonetics, and morphology of each language independently. Accordingly, the manner of articulation constrains the possible phonemic/morphemic combinations in a language. For example in Arabic, it is hard to find a word that have the "th" sound followed by an "s" sound, while it is possible in English as in the word "thus". Historically, the famous Arab lexicographer Al-Farahidi (718 - 786 CE) noticed this phenomenon (where certain sound sequences are allowed while others are not in the language) and devised a method by which he can distinguish Arabic words from foreign ones on the basis of the possible sequences of letters in Arabic (Ahmad, 2003). Though having closed set of impossible sequences of letters in each language could help in distinguishing between languages within an utterance, but in reality it is hard to find such sets for all languages. Hence, we believe that building a character level n-gram language model for the target language to maximize the probabilities of the possible patterns and suppress the probabilities of the impossible ones should provide an approximation to such a closed set rule-based list. Not to mention that producing such a list by hand is quite laborious and error prone as a process.

Accordingly, we propose a supervised learning framework to address the challenge of LCSPD. We assume the presence of annotated code switched training data where each token is annotated as either Lang1 or Lang2. We create a sequence model using Conditional Random Fields (CRF++) tool (Sha and Pereira, 2003). For each word in the training data, we create a feature vector comprising word length, character sequence level probabilities, and unigram word level probabilities. Once we derive the learning model, we apply to input text to identify Lang1 tokens vs. Lang2 tokens in context. For the character sequence level probabilities, we build (1, 2, 3, 4, and 5)-gram character language models (CLMs) using the SRILM tool (Stolcke, 2002) for each of the two languages presented in the training data using the annotated words. For example, if the training data contains the two languages "lang1" and "lang2", we use all words that have the "lang1" tags to build (1, 2, 3, 4, and 5)-grams CLMs for "lang1" and the same for "lang2". We apply all of the created CLMs to each word in the training data to find their character sequence probabilities in each language in the training data. To increase the difference between the feature vectors of the words related to "lang1" and those related to "lang2", we use a word level unigram LM for each of the two languages in the training data. Then we apply the unigram LMs to each word in the training data to find their word level probabilities in each language in the training data, i.e. checking whether it pertains to the language or not by virtue of having a higher probability in the corresponding LM than words not in the language.

4 Experimental Setup

4.1 Data

We evaluate our proposed framework on different language pairs exhibiting code switching. We use the training and test data sets provided by the shared task for "Language Identification in Code-Switched Data" [ShTk] in 2014 and 2016 (Solorio et al., 2014; Molina et al., 2016). The ShTk-2014 datasets includes English-Spanish, English-Nepali, Modern standard Arabic (MSA)-Egyptian Arabic (EGY), and English-Mandarin[3], while the ShTk-2016 datasets includes English-Spanish, and, MSA-EGY. In addition to these languages, we evaluate our system on MSA-Levantine (LEV), MSA-Gulf, Arabizi-English, Arabic-Engari, and English-Hindi datasets[4].

[3]Unfortunately, we did not manage to get English-Mandarin datasets from the organizers but we got the rest of them.

[4]Nepali, Arabizi, and Hindi are written using Roman script. Engari is written using Arabic script

- MSA-LEV and MSA-Gulf: These datasets are collected from online newspaper commentary and Twitter by Cotterell and Callison-Burch (2014). The datasets are annotated for sentence level. We re-annotated the data for token level using the guidelines provided by Diab et al. (2016). We then split the data into training and test (80% for training and 20% for testing);

- Arabizi-English: We use the same training and test sets used by 3ARRIB. The data sets are created by the Linguistic Data Consortium from SMS/Chat corpus (Bies et al., 2014; LDC, 2014a; LDC, 2014b; LDC, 2014c);

- Arabic-Engari: Same as the MSA-EGY data sets. But we re-annotated the data to tag all English words that are written in Arabic script. MSA and EGY words are both tagged as Arabic words;

- English-Hindi: It consists of 728 and 376 sentences for training and test sets, respectively, collected from Twitter and Facebook. This dataset is part of a corpus that is used for POS-tagging experiment in code-switched data (Jamatia et al., 2015).

Table 1 shows the distribution of each language in the training and test sets. The lang1, lang2 labels refer to the two languages addressed in the dataset name, for example for the language pair English-Spanish, lang1 is English and lang2 is Spanish, in that order[5].

All Language-Pairs	Training-Set		Test-Set	
	lang1	lang2	lang1	lang2
English-Spanish-2014	77,101	33,099	7,424	5,278
Nepali-English-2014	60,493	44,111	12,286	17,216
MSA-EGY-2014	79,059	16,291	57,740	21,871
Arabizi-English-2014	93,402	11,122	27,308	1,903
Arabic-Engari	439,875	1,282	79,611	433
English-Hindi	6,562	5,526	8,676	378
LEV-MSA	44,694	11,522	11,524	2,265
Gulf-MSA	57,718	8,655	15,400	1,409
English-Spanish-2016	77,101	33,099	32,442	123,973
MSA-EGY-2016	79,059	16,291	5,804	9,630

Table 1: Language distribution (words/language) in the training and test data sets for all language-pairs

In addition to the training data described in table 1, we used the following datasets to improve the word level LMs of the English, Spanish and Arabic langauges:

- English Gigaword (LDC, 2003b): To build the unigram word level LM for the English part in English-Spanish, English-Nepali, and English-Hindi language-pairs;

- Spanish Gigaword (LDC, 2009): To build the unigram word level LM for the Spanish part in English-Spanish language-pair;

- Arabic Gigaword (LDC, 2003a): To build the unigram word level LM for the MSA part in MSA-EGY, LEV-MSA, and Gulf-MSA language-pairs;

- Egyptian discussion forums (LDC, 2012): To build the unigram word level LM for the EGY part in MSA-EGY language-pairs. It is also used in addition to the Arabic Gigaword to build the LM for the Arabic part in the Arabic-Engari language pair.

4.2 Baselines

We evaluate our approach against the best published results using the same training and test sets. The baselines include:

[5]The ShTk-2014 has a different naming convention for the Nepali-English, however we opt for changing the naming to indicate the majority class is Nepali as in lang1 and English is the minority class language lang2

- Majority: In this baseline, for each word in the test set, we check the most frequent tag for that word in the training set and assign it to that word. If the word is not in the training set, we give it the most frequent language tag observed overall in the training data;
- TAU: The results on the English-Spanish dataset obtained by Bar and Dershowitz (2014);
- DCU-UVT: The results on the English-Spanish and English-Nepali datasets obtained by Barman et al. (2014);
- CMU: The results on the English-Spanish, English-Nepali, and MSA-EGY datasets obtained by Lin et al. (2014);
- IUCL: The results on the English-Spanish, English-Nepali, and MSA-EGY datasets obtained by King et al. (2014);
- IIIT: The results on the English-Spanish, English-Nepali, and MSA-EGY datasets obtained by Jain and Bhat (2014);
- MSR-India: The results on the English-Spanish, English-Nepali, and MSA-EGY datasets obtained by Chittaranjan et al. (2014);
- AIDA: The results on the MSA-EGY dataset obtained by our contribution in the ShTk-2014 (Elfardy et al., 2014);
- AIDA2: The results on the MSA-EGY dataset obtained by our previous publication about AIDA2 system (Al-Badrashiny et al., 2015);
- 3ARRIB: The results on the Arabizi-English dataset obtained by Eskander et al. (2014);
- IIIT Hyderabad and HHU-UH-G: The best results on the English-Spanish-2016 and MSA-EGY-2016 respectively[6].

5 Evaluation

Table 2 summarizes the published results by all baselines systems on the English-Spanish-2014, English-Nepali-2014, Arabizi-English, and MSA-EGY-2014 datasets. The table shows that TAU is the best published system on the English-Spanish-2014 data, DCU-UVT is the best published system on the English-Nepali-2014 data, 3ARRIB is the best published system on the Arabizi-English, and AIDA2 is the best published system on the MSA-EGY-2014 data.

Table 3 shows the results of our system on all language-pairs compared to the best published results from table 2 and the majority baseline. Lang1 indicates the majority class as per the training data, while lang2 indicates the minority class in the training data. The results show that LILI yields competitive results compared to all published state-of-the-art systems. The overall weighted average F-score for LILI is higher than all the majority baselines. It is also either higher than the published state-of-the-art systems except in the MSA-EGY compared to AIDA2 and HHU-UH-G, or very close to the best published results as in the English-Spanish-2016 dataset (LILI got 95.2% compared to 96% of IIIT Hyderabad with only 0.8% difference). Arabic language pairs; MSA-EGY, Gulf-MSA, and LEV-MSA are the most challenging ones. Because unlike the other languages, the words in each of these pairs do not create disjoint sets, as mentioned earlier, there is significant overlap hence they share significant character and word patterns. This issue is even worse because the native Arabic speakers do not write the short vowels (also know as diacritics) while they are able to reconstruct them while reading without any problem. The MSA shares many words with the other Arabic dialects but with different diacritics. For example, the words (yalEabuwn) in MSA and (yilEabuwn) in Gulf (Both mean they are playing)

[6]We are unaware of the citations of these 2 papers since they are not published yet while writing this paper. We got the systems names and their results from the shared task website `http://care4lang1.seas.gwu.edu/cs2/call.html`

Language-Pairs	System	lang1	lang2	Avg-F
English-Spanish-2014	CMU	93.30%	93.60%	93.42%
English-Spanish-2014	DCU-UVT	93.60%	92.70%	93.23%
English-Spanish-2014	**TAU**	**95.20%**	**95.20%**	**95.20%**
English-Spanish-2014	IUCL	94.10%	93.20%	93.73%
English-Spanish-2014	IIIT	92.90%	92.00%	92.53%
English-Spanish-2014	MSR-India	94.20%	93.80%	94.03%
English-Nepali-2014	CMU	91.40%	93.20%	92.45%
English-Nepali-2014	**DCU-UVT**	**97.40%**	**96.50%**	**96.87%**
English-Nepali-2014	IUCL	80.80%	87.10%	84.48%
English-Nepali-2014	IIIT	96.90%	94.30%	95.38%
English-Nepali-2014	MSR-India	96.90%	94.80%	95.67%
Arabizi-English	**3ARRIB**	**97.40%**	**75.80%**	**95.99%**
MSA-EGY-2014	CMU	89.90%	81.10%	87.48%
MSA-EGY-2014	IUCL	81.10%	59.50%	75.17%
MSA-EGY-2014	IIIT	86.20%	52.90%	77.05%
MSA-EGY-2014	MSR-India	86.00%	56.40%	77.87%
MSA-EGY-2014	AIDA	89.40%	76.00%	85.72%
MSA-EGY-2014	**AIDA2**	**92.90%**	**82.90%**	**90.15%**

Table 2: Summary results of all published systems that use English-Spanish-2014, English-Nepali-2014, Arabizi-English, and MSA-EGY-2014 datasets. For each group, the F-score is presented for lang1 and lang2 followed by the weighted average F-score for both languages.

All	LILI			Best Published Results				Majority Baseline		
Language-Pairs	lang1	lang2	Avg-F	System	lang1	lang2	Avg-F	lang1	lang2	Avg-F
English-Spanish-2014	97.2%	95.3%	**96.4%**	TAU	95.2%	95.2%	95.2%	92.8%	88.0%	90.8%
Nepali-English-2014	97.6%	97.0%	**97.3%**	DCU-UVT	97.4%	96.5%	96.9%	92.8%	94.0%	93.3%
English-Hindi	98.8%	80.4%	**98.0%**	NA	NA	NA	NA	98.4%	64.9%	97.0%
Arabizi-English	98.3%	77.9%	**97.0%**	3ARRIB	97.4%	75.8%	96.0%	86.9%	36.5%	83.6%
Arabic-Engari	99.1%	64.2%	**98.9%**	NA	NA	NA	NA	95.0%	62.4%	94.8%
MSA-EGY-2014	86.0%	87.2%	86.3%	AIDA2	92.9%	82.9%	**90.2%**	70.9%	63.7%	68.9%
LEV-MSA	93.6%	61.0%	**88.2%**	NA	NA	NA	NA	90.1%	25.1%	79.4%
Gulf-MSA	95.5%	36.6%	**90.6%**	NA	NA	NA	NA	94.6%	13.6%	87.8%
English-Spanish-2016	88.6%	96.9%	95.2%	IIIT Hyderabad	92.3%	96.9%	**96.0%**	77.4%	84.1%	82.7%
MSA-EGY-2016	82.0%	86.8%	85.0%	HHU-UH-G	85.4%	90.4%	**88.5%**	59.6%	47.4%	52.0%

Table 3: Summary results of our system performance on all language-pairs compared to the best published results and Majority baselines. For each group, the F-score is presented for lang1 and lang2 followed by the weighted average F-score for both languages. There are no published systems for English Hindi, Arabic-Engari, LEV-MSA, and Gulf-MSA hence the NA (not available).

become the same after removing the short vowels (ylEbwn). Hence, modeling more nuanced features is needed such as POS tags and morphological information, which is the case in the AIDA2 system. But despite the simplicity of the presented approach in this paper, in the MSA-EGY-2014 and MSA-EGY-2016 datasets, our yielded weighted average F-scores (87.2% and 85.0%) are not far from the AIDA2 (90.15%) and HHU-UH-G (88.5%) scores. Furthermore, It worth mentioning that running AIDA2 on the MSA-EGY-2016 gives 85.2% weighted average F-score; which is lower than LILI.

In a supervised framework, minority class detection is the more challenging task. We compare our performance to the published systems where available as well as the majority baseline. Our results are comparable to the published state-of-the-art systems, even significantly better in the MSA-EGY-2014 dataset. Moreover, our results outperform the majority baseline in all language pairs.

We can notice from table 2 that neither of the baselines published systems achieve the best results across the different language pairs. For example the DCU-UVT system achieved best result on Nepali-EN-2014 but it did not even achieve the second best on Spanish-EN-2014. Although the MSR-India has higher result than the DCU-UVT on the Spanish-EN-2014, it has lower results than it on the Nepali-EN-2014 data. This shows that despite the simplicity of our approach, it is generic and outperforms the states of the art systems or competitively compared to them across all the language pairs.

The above results show that the proposed approach is working well on the binary problems when we know beforehand the word is either lang1 or lang2. However, in real code switching scenario we do not know what the other language variety could be in the data. To see how robust our approach is, we tested our approach on a multi-class model. We trained a single multi-class CRF model using a combined data from our different training sets and tested it on the corresponding combined test sets. Unfortunately, we didn't manage to create a CRF model using all our training data. The datasets we managed to use are the English-Spanish-2014, Nepali-English-2014, English-Hindi, Arabizi-English, and MSA-EGY-2014. Table 4 shows the F-score results of our system on all languages using the multi-class model. We didn't find any published systems that conducted the same experiment to compare our results to. Therefore, table 4 only compares our results to the majority baseline and the binary model in table 3. The results show that LILI outperforms the majority baseline on all languages. The F-scores on all languages are comparable to the F-scores from the binary-classes case except on Hindi, where the multi-class model is much lower than the binary one. However, the overall weighted average F-score is high (91.0%). This shows that LILI is able to perform in a good way on the real code switching problem.

Language	LILI-Multi-Class-Model	LILI-Binary-Model	Majority Baseline
Nepali	97.4%	97.6%	90.0%
Hindi	62.3%	80.4%	30.7%
Arabizi	95.6%	98.3%	80.7%
English	96.2%	96.4%	93.4%
MSA	85.5%	86.0%	70.0%
Spanish	94.3%	95.3%	79.1%
Engari	64.0%	64.2%	56.5%
EGY	87.7%	87.2%	63.0%
Avg-F	91.0%	91.2%	77.8%

Table 4: The F-score of our system compared to the majority baseline on all languages using a single multi-class model. The last row shows the weighted average F-score for all languages.The number in the English row of LILI-Multi-Class-Model is the weighted average F-score of the English label obtained by LILI in table 3 on English-Spanish-2014, Nepali-English-2014, English-Hindi, and Arabizi-English datasets.

Finally, the simplicity of our system made it very fast. It can process up to 20,000 words/sec; which renders it very efficient and amenable to large scale processing. We compare our system's speed to our previously published tools; AIDA2 and 3ARRIB. AIDA2 processes 1000 words/sec and 3ARRIB processes 49 words/sec. Hence LILI is orders of magnitude faster than both systems with a relatively minor drop in performance compared to AIDA2 if we consider overall F1-score, or better performance if we care about detecting the minority class, and better performance than 3ARRIB overall.

6 Conclusion

In this paper, we introduced a simple yet powerful framework for the linguistic code switch point detection problem. The solution is language independent, thus it can work with any language-pair. The framework is based on the idea that each language has its own phonological system that control which set of sounds can occur together. Our assumption was that this is sufficient to distinguish between languages used in the same utterance. The results show that our simple approach outperforms or at least performs competitively compared to state-of-the-art complex machinery systems when evaluated against standard data sets with the added advantage of speeds that could be scaled up to big data levels.

References

Al-Khalil Ibn Ahmad. 2003. *Kitab al-?Ain*. Edited by Abdulhameed Hindawy(2003), Dar Al-kotob Alilmiyah, Beirut, Lebanon, first edition.

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic Transliteration of

Romanized Dialectal Arabic. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Mohamed Al-Badrashiny, Heba Elfardy, and Mona Diab. 2015. Aida2: A hybrid approach for token and sentence level dialect identification in arabic. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 42–51, Beijing, China, July. Association for Computational Linguistics.

Waleed Ammar, Chris Dyer, and Noah A Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3311–3319. Curran Associates, Inc.

Kfir Bar and Nachum Dershowitz. 2014. The tel aviv university system for the code-switching workshop shared task. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 139–143, Doha, Qatar, October. Association for Computational Linguistics.

Utsab Barman, Joachim Wagner, Grzegorz Chrupała, and Jennifer Foster. 2014. Dcu-uvt: Word-level language classification with code-mixed data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 127–132, Doha, Qatar, October. Association for Computational Linguistics.

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rabmow. 2014. Transliteration of Arabizi into Arabic Orthography: Developing a Parallel Annotated Arabizi-Arabic Script SMS/Chat Corpus. In *Arabic Natural Language Processing Workshop, EMNLP*, Doha, Qatar.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury, 2014. *Proceedings of the First Workshop on Computational Approaches to Code Switching*, chapter Word-level Language Identification using CRF: Code-switching Shared Task Report of MSR India System, pages 73–79. Association for Computational Linguistics.

Ryan Cotterell and Chris Callison-Burch. 2014. A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic. In *LREC*.

Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Nada AlMarwani, and Mohamed Al-Badrashiny. 2016. Creating a large multi-layered representational repository of linguistic code switched arabic data. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab, 2014. *AIDA: Identifying Code Switching in Informal Arabic Text*, chapter Proceedings of the First Workshop on Computational Approaches to Code Switching, pages 94–101. Association for Computational Linguistics.

Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. 2014. Foreign words and the automatic processing of arabic social media text written in roman script. *In Proceedings of the First Workshop on Computational Approaches to Code-Switching. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing, October, 2014, Doha, Qatar*.

Naman Jain and Ahmad Riyaz Bhat, 2014. *Proceedings of the First Workshop on Computational Approaches to Code Switching*, chapter Language Identification in Code-Switching Scenario, pages 87–93. Association for Computational Linguistics.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 239–248, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Levi King, Eric Baucom, Timur Gilmanov, Sandra Kübler, Daniel Whyatt, Wolfgang Maier, and Paul Rodrigues. 2014. The iucl+ system: Word-level language identification via extended markov models. In *Proceedings of the First Workshop on Computational Approaches to Code-Switching*. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing, Doha, Qatar.

LDC. 2003a. Arabic Gigaword Fifth Edition LDC2011T11. Linguistic Data Consortium.

LDC. 2003b. English Gigaword LDC2003T05. Linguistic Data Consortium.

LDC. 2009. Spanish Gigaword Second Edition LDC2009T21. Linguistic Data Consortium.

LDC. 2012. BOLT - Phase 1 Discussion Forums Source Data R4 LDC2012E54. Linguistic Data Consortium.

LDC. 2014a. BOLT Phase 2 SMS and Chat Arabic DevTest Data – Source Annotation, Transliteration and Translation. LDC catalog number LDC2014E28.

LDC. 2014b. BOLT Phase 2 SMS and Chat Arabic Training Data – Source Annotation, Transliteration and Translation R1. LDC catalog number LDC2014E48.

LDC. 2014c. BOLT Program: Romanized Arabic (Arabizi) to Arabic Transliteration and Normalization Guidelines. Version 3. Linguistic Data Consortium.

Chu-Cheng Lin, Waleed Ammar, Lori Levin, and Chris Dyer, 2014. *Proceedings of the First Workshop on Computational Approaches to Code Switching*, chapter The CMU Submission for the Shared Task on Language Identification in Code-Switched Data, pages 80–86. Association for Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of The EMNLP 2016 Second Workshop on Computational Approaches to Linguistic Code Switching (CALCS)*.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M. Roth. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of LREC*, Reykjavik, Iceland.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL*, pages 213–220, Edmonton, Canada.

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, October, 2014, Doha, Qatar*.

Andreas Stolcke. 2002. Srilm an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*.