POS Tagging for Improving Code-Switching Identification in Arabic

Mohammed Attia[1] **Younes Samih**[2] **Ali Elkahky**[1] **Hamdy Mubarak**[2]
Ahmed Abdelali[2] **Kareem Darwish**[2]
[1]Google LLC, New York City, USA
[2]Qatar Computing Research Institute, HBKU Research Complex, Doha, Qatar
[1]{attia,alielkahky}@google.com
[2]{ysamih,hmubarak,aabdelali,kdarwish}@hbku.edu.qa

Abstract

When speakers code-switch between their native language and a second language or language variant, they follow a syntactic pattern where words and phrases from the embedded language are inserted into the matrix language. This paper explores the possibility of utilizing this pattern in improving code-switching identification between Modern Standard Arabic (MSA) and Egyptian Arabic (EA). We try to answer the question of how strong is the POS signal in word-level code-switching identification. We build a deep learning model enriched with linguistic features (including POS tags) that outperforms the state-of-the-art results by 1.9% on the development set and 1.0% on the test set. We also show that in intra-sentential code-switching, the selection of lexical items is constrained by POS categories, where function words tend to come more often from the dialectal language while the majority of content words come from the standard language.

1 Introduction

Code-switching (CS) is common in multilingual communities as well as diglossic ones, where the language of information and education is different from the language of speaking and daily interaction. With the increased level of education, mobility, globalization, multiculturalism, and multilingualism in modern societies, combined with the rise of social media, where people write in the way they speak, CS has become a pervasive phenomenon, particularly in user-generated data, and a major challenge for NLP systems dealing with that data.

CS is interesting for two reasons: first, there is a large population of bilingual and diglossic speakers, or at least speakers with some exposure to a foreign language, who tend to mix and blend two languages for various pragmatic, psycholinguistic and sociolinguistic reasons. Second, existing theoretical and computational linguistic models are based on monolingual data and cannot adequately explain or deal with the influx of CS data whether spoken or written.

CS has been studied for over half a century from different perspectives, including theoretical linguistics (Muysken, 1995; Parkin, 1974), applied linguistics (Walsh, 1969; Boztepe, 2003; Setati, 1998), socio-linguistics (Barker, 1972; Heller, 2010), psycho-linguistics (Grosjean, 1989; Prior and Gollan, 2011; Kecskes, 2006), and more recently computational linguistics (Solorio and Liu, 2008a; Çetinoğlu et al., 2016; Adel et al., 2013b).

In this paper, we investigate the possibility of using POS tagging to improve word-level language identification for diglossic Arabic in a deep-learning system. We present some syntactic characterization of intra-sentential code-switching, and show that POS can be a powerful signal for code-switching identification. We also pay special attention to intra-sentential code-switching and examine the distribution of POS categories involved in this type of data.

The paper is organized as follows: in the remainder of this introduction we present challenges, definitions, and types of CS, and the particular aspects involved in Arabic CS. Section 2 gives an overview of related works. In Section 3, we describe and record our observations on the data used in our experiments. Section 4 presents a description of our system and the features used. Section 5 gives the details of our experiments and discusses the results, and finally we conclude in Section 6.

1.1 Why is CS Computationally Challenging?

When two languages are blended together in a single utterance, the traditional phonological and morphosyntactic rules are perturbed. When

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 18–29
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

judged by a standard monolingual model, these utterances can be deemed as ungrammatical or unnatural. Therefore, CS should generally be treated in its own terms and not to be conceived of as a peripheral phenomenon that can be understood by tweaking and twisting monolingual models and theories. When two languages come in contact, this implies the cross-fertilization and the emergence of structures that may be absent in either languages. When code-switching, speakers compromise the syntactic rules of the two languages involved, sometime adding in or leaving out a determiner, or applying a system of affixation from one language and not the other.

CS has conveniently been used as a cover term (Myers-Scotton, 1997; Çetinoğlu et al., 2016) for all operations where two languages are used simultaneously or alternately by the same speaker. When the user speaks one sentence in one language and another sentence in another language, this has been referred to as inter-sentential code-switching, while mixing elements from the two languages together in the same sentence has been termed intra-sentential. The language that provides the function words and grammatical structure is called the host (Bokamba, 1989) or matrix language, while the language being inserted is called the guest or embedded language.

While inter-sentential CS is relatively less challenging for computational analysis, as each sentence still follows a monolingual model, intra-sentential CS poses a bottleneck challenge. It needs a special amount of attention, because it is only this type that involves the lexical and syntactic integration and activation of two language models at the same time. NLP systems trained on monolingual data suffer significantly when trying to process this kind bilingual text or utterance.

CS has proved challenging for NLP technologies, not only because current tools are geared toward the processing of one language at a time (AlGhamdi et al., 2016), but also because code-switched data is typically associated with additional challenges such as the non-conventional orthography, non-canonicity (nonstandard or incomplete) of syntactic structures, and the large number of OOV-words (Çetinoğlu et al., 2016), which suggest the need for larger training data than what is typically used in monolingual models. Unfortunately, shortage of training data has usually been cited as the reason for the under-performance of computational models when dealing with CS data (Adel et al., 2015).

The study of CS does not only help downstream tasks (like ASR (automatic speech recognition), IR (information retrieval), parsing, etc.), but it is also crucial for language generation (e.g. TTS (text to speech), MT (machine translation), and automated responses by virtual assistants) in order to allow computational models to produce natural sentences that closely match how modern societies talk.

1.2 Definition and Defining Perspectives

The definition of CS has varied greatly depending on the different researchers' attitude and perspectives of the operation involved. While some viewed it as a process where two languages are actively interacting with each other (ultimately creating a new code), other viewed the operation just as two separate languages sitting side-by-side as isolated islands. Following the first perspective, Joshi (1982) defined code-switching as the situation when two languages systematically interact with each other in the production of sentences in a framework which consists of two grammatical systems and a mechanism for switching between the two. Following the second perspective, Muysken (1995) defined CS as "the alternative use by bilinguals of two or more languages in the same conversation", while other researchers (Auer, 1999; Nilep, 2006) defined it as the "juxtaposition" of elements from two different grammatical systems within the same speech.

The *juxtaposition* definition has been widely cited in the research on code-switching, advancing a monolingual view on the topic and promoting the idea that bilingual speech is the sum (or juxtaposition) of two monolingual utterances. The literal meaning suggests placing two heterogeneous and isolated pieces from different languages next to each other, but, in fact, foreign phrases are usually syntactically integrated and may often change phonologically, morphologically and pragmatically to fit homogeneously in the new position. The term also has a sense of randomness, which departs from the fact that CS is patterned and predictable.

The view we adopt is that when people code-switch, they interweave (Lipski, 2005) or blend two languages together, and the grammar of code-switching depends, to a large extent, on which lan-

guages are being interwoven, where, when, how, and by whom. The *where* and *when* relates to the sociolinguistic factors, such as the situation and power relations, and the *how* and *by whom* to the psycholinguistic factors, such as speakers' competence and proficiency in either or both languages. This is why we see a wide range of regular patterns as well as highly idiosyncratic behavior.

1.3 CS Types and Categories

A speaker can turn from one language to the other at the sentence level, or he/she can make the turn within the same sentence. Some researchers (Muysken et al., 2000) use the term "code-switching" to refer to the former case while reserving the term "code-mixing" to refer to the latter. However, these two types have more conventionally been termed as inter-sentential and intra-sentential code-switching, respectively, as explained above.

Intra-sentential CS has further been divided by Muysken et al. (2000) into three types: 1) insertion where words or phrases from one language are inserted into another, 2) alternation where there is a total shift from one language into the other, e.g. starting the sentence in one language and ending in another, and 3) congruent lexicalization similar to insertion, but with a high frequency, and found in typologically similar language pairs by fluent bilinguals.

Another classification is by looking at the nature of the language pairs, CS can be classified as diglossic, i.e. between varieties of the same language (e.g. Standard and Egyptian Arabic); typologically-related, i.e. between language pairs that belong to the same language family (e.g. English and Spanish); or typologically-distinct, i.e. between language pairs that come from different language families (e.g. Chinese and English). It has been suggested that CS between typologically similar languages is facilitated in ways that are different from (and not found in) those in typologically distinct languages (Lipski, 2005; Chan, 2009). By contrast, dialect/standard variation has been viewed by some as a form of style shifting (Trudgill, 1986) rather than proper CS, while others argue that style-shifting may serve the same kind of functions in conversation as CS (Boztepe, 2003), and that CS can happen between language varieties as well as different languages (Gardner-Chloros, 1991). It is to be noted however, that in diglossic code-switching, the shift is more likely to be lexical, morphological, and structural, rather than phonological, unlike the other two cases when we have two completely distinct language systems.

1.4 Peculiarities of Arabic CS

Arabic is a diglossic language, where the language of education is different from the language of speaking. Dialectal Arabic has traditionally not enjoyed the same prestige, socio-economic status, and official recognition as MSA. Dialects, by nature, diverge from the standard language, and, therefore, they can easily and freely draw from the larger repository of the standard language.

It has been suggested that CS most frequently happens from the subordinate language to the more superior one not vice versa (Lipski, 2005). This, however, might be true in general, but not in the absolute sense, as CS to the so-called subordinate language may be for the back-stage communicative purposes (e.g. establishing identity and friendliness or referencing a cultral meme).

Code-switching to MSA is used to establish authority and maintain credibility. Using the dialect (or mother tongue) on the other hand signifies a sense of belonging, community and solidarity, and attracts a higher level of attention and understandability. In other words, MSA is the intellectual language, while dialect is the emotive one.

The data used in the experiments in this paper comes from Twitter which are in the written modality, and this can significantly vary from the spoken interactions. Arabic speakers' competence in spoken MSA is remarkably lower than in the written one. While most Arabic speakers with some level of education can write in MSA, far fewer are able to utilize MSA in speaking. Spoken CS can be observed more with public speakers, like presenters, politicians and lecturers, and less often with ordinary people.

Moreover, there is a large number of lexical items which have shared orthography in EA (Egyptian) and MSA, though the pronunciation is different, e.g. متأكد muta>ak~id/mito>ak~id "sure", كاملة kamilap/kamolap "full", and قريب qariyb/quray~ib "near". This is generic to some extent, as the pattern mutaR$_1$aR$_2$~iR$_3$, for instance, is changed to mitoR$_1$aR$_2$~iR$_3$ where R stands for the root letter, or cardinal. As Twitter data is written without diacritization, there is

no way to know precisely whether words are pronounced with dialectal or standard accent, though the context can give some clue, and we think that this kind of distinction was left to the annotators' best judgment.

Arabic, as a morphologically-rich language, has its peculiar behavior of merging morphemes and clitics from the matrix language to the embedded language. In diglossic mixed codes, standard verbs can show dialectal morphology, whether through affixes or templatic vowel shifting, e.g. هيرسلهالهم hayirosilhAlohum "will send it to them". For foreign words, they can receive agreement morphology هيكبيها haykabiyhA "he will copy it". This type of morpho-syntactic blending is stereotypical of CS when Arabic, or one if its dialects, is the matrix language.

2 Related Work

2.1 Computational Approaches

Research on computational approaches to CS has been mainly concentrated in four areas: predicting code-switching points, word-level language identification, POS tagging, and automatic speech recognition. However, some relatively recent research has tried to tackle CS in MT (Johnson et al., 2017), question answering (Raghavi et al., 2015), sentiment analysis (Vilares et al., 2015) and information retrieval (Chakma and Das, 2016).

The task of predicting code-switching points is significantly different from word-level code-switching identification, because in the former the classifier is allowed only to look at the past (previous) words and predict which language the coming word is going to be in, whereas in the latter, the classifier has the fuller context and evidently can achieve much higher accuracy. Moreover the former focuses on the elements or points after which you can make the switch, while the latter looks at the elements being switched themselves.

Solorio and Liu (2008a) pioneered the work on CS and developed an ML (machine learning) classifier to predict code-switching points in Spanish-English. The data they used was recorded conversations among three English-Spanish bilingual speakers. The conversations included 922 sentences and were manually transcribed and annotated with POS tags. They trained their Naive Bayes classifier on a number of features including language ID, lemma and POS tags and reported an f-score of 28%, with 1% positive variance gained through the POS feature.

In another effort, Solorio and Liu (2008b) tried POS tagging on Spanish-English CS data and concluded that feeding the output of two monolingual taggers to an ML algorithm yielded the best results.

Çetinoğlu et al. (2016) pointed out that POS tagging of CS data proved much harder than tagging monolingual texts, as models could reach 97% accuracy for the latter, but only around 77% for the former. They attribute the poor performance largely to the lack of CS annotated data, and the fact that many systems just devise methods to choose from the output of two monolingual POS taggers, e.g. the work of Solorio and Liu (2008b) and Sharma et al. (2016).

Similar to the work of (Solorio and Liu, 2008a), Adel et al. (2013b,a) tried to predict code-switching points for conversational speech in the Mandarin-English SEAME corpus to improve an ASR model. They used recurrent neural network language modeling relying on POS tags and using a factorized output layer. They noted that speakers most frequently switch to another language for nouns and object noun phrases. They also assumed that the switching attitude is speaker-dependent and clustered speakers into classes with similar switching attitude. They reported an accuracy of 43.31% and proved that POS tags have statistically significant role on improving the results. Adel et al. (2013b) tried to accommodate bilingual data by merging monolingual resources, such as the English and Mandarin Dictionaries, the output of two separate POS taggers, the Stanford POS tagger for Mandarin, and the Stanford tagger for English, and using two monolingual language models. Additionally they hard-coded some phonological rules to accommodate Singaporean English. They later extended their features to include Brown clusters, open class words and word embeddings (Adel et al., 2015) and found that Brown word clusters, part-of-speech tags and open-class words are the most effective at reducing the perplexity.

Fewer studies have focused on CS between related language varieties which is typically a diglossic kind of CS between a standard language and a dialect, e.g. Cypriot Greek and Standard Modern Greek (Tsiplakou, 2009).

CS research on Arabic included POS tagging and word-level language identification. AlGhamdi

et al. (2016) explored different technique for the POS tagging of CS data and concluded that applying a machine learning framework as a voting mechanism on top of the output of two monolingual POS taggers achieves the best performance. Word-level CS identification for Arabic (along with Spanish–English) has been featured in a couple of shared tasks: the First Shared Task on Language Identification in Code-Switched Data (Solorio et al., 2014) and the Second Shared Task on Language Identification in Code-Switched Data (Molina et al., 2016), of which Samih et al. (2016) was the winning system, and against which we compare our results in this project.

Eskander et al. (2014) studied CS between EA written in Roman script (Arabizi) and English. Habash et al. (2008) created a standard annotation guidelines for CS between MSA and dialects.

CS has also been studied in Arabic as a predictor of social influence in the collaborative writing in Wikipedia discussion pages in (Yoder et al., 2017) and it was found that CS is positively associated with the editor's success in winning an argument.

We notice from the literature that in some instances POS tagging has been used to aid with the identification of code-switching points, and in some other instances language identification has been used as an indicator or a feature for POS tagging, showing what (Çetinoğlu et al., 2016) referred to as task inter-relatedness, or the cyclic nature of task dependencies. In our work, we use a POS tagger as a predictor of CS. The POS tagger used has been trained specifically on CS data.

3 Data Description

The organizers of the Second Shared Task on Language Identification in Code-Switched Data (Molina et al., 2016) provided the annotated dataset for the MSA–EA code-switched pairs. The data consists of 8,862 tweets (185,928 tokens) as training set, 1,117 tweets (20,688 tokens) as development set and 1,262 tweets (20,713 tokens) as final test set. The tagset statistics for the training set are shown in Table 1.

Furthermore, the training data contains 970 (11%) intra-sentential CS tweets, i.e. tweets with both $lang1$ (MSA) and $lang2$ (EA); 865 (10%) tweets with $lang2$ only; and the remaining tweets (79%) with $lang1$ only.

We analyze the POS distribution in the data using the prediction of a specially designed POS tagger, described in 4.1, and notice that in those intra-sentential CS sentences, the majority of function words (particles, adverbs and pronouns) come from lang2 (dialect), while the majority of content words (adjectives, verbs and nouns) come from the $lang1$ (standard language). The distribution of $lang1$ and $lang2$ by POS is shown in Figure 1.

Labels	Token Count	Token Ratio %
ambiguous	1,186	0.64
unk	0	0.00
lang1	127,690	68.70
lang2	21,722	11.69
mixed	16	0.01
ne	21,567	11.60
other	13,691	7.37

Table 1: Tag count and ratio in the training set, where $lang1$ is MSA, $lang2$ is EA, and ne is a named entity.

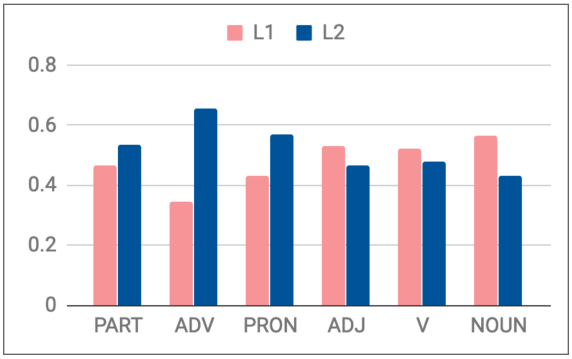

Figure 1: POS Distribution in CS data

Figure 2 shows CS behavior on a sample of users, and it indicates that the switching attitude is idiosyncratic and user-dependent.

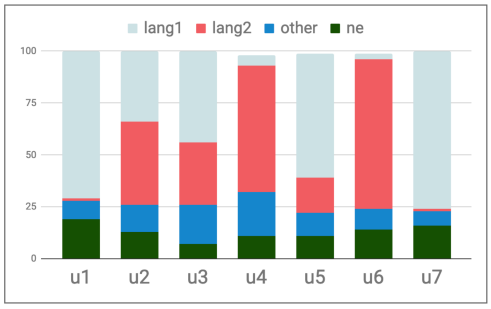

Figure 2: CS Distribution by Users

Data preprocessing: We transformed Arabic scripts to SafeBuckwalter (Roth et al., 2008), a character-to-character mapping that replaces Arabic UTF alphabet with Latin characters to reduce

size and streamline processing. Also in order to reduce data sparsity, we converted all Persian numbers (e.g. ۲ ،۱) to Arabic numbers (e.g. 1, 2), Arabic punctuation (e.g. '،' and '؛') to Latin punctuation (e.g. ',' and ';'), removed kashida (elongation character) and diacritics, and separated punctuation marks from words.

4 System Description

Deep learning and neural nets have been used extensively in the past decade and were shown to significantly outperform traditional (linear) ML models. The proclaimed advantage of deep learning is that it eliminates the need for feature engineering. Yet, there has been a growing interest recently to augment neural nets with more and more linguistic features, which has been shown to boost performance for many tasks.

We use a DNN (Deep Neural Network) model mainly suited for sequence tagging and is a variant of the bi-LSTM-CRF architecture (Ma and Hovy, 2016; Lample et al., 2016; Reimers and Gurevych, 2017; Huang et al., 2015). Our implementation is mostly inspired by the work of Reimers and Gurevych (2017). In its basic configuration, it combines a double representation of the input words by using word embeddings and a character-based representation with CNNs (convolutional Neural Networks). The input sequence is processed with bi-LSTMs, and the output layer is a linear chain CRF. We augment this model with various layers to accommodate the different features we want to incorporate. The features used in our model are explained below.

4.1 Dialectal POS Tagger

We develop a POS tagger using the data described in Darwish et al. (2018). The tagger used in this paper is developed using a deep neural network model, unlike Darwish et al. (2018) who use a linear model. Our model predicts POS tagging at the word level (not the token level), to suit how the CS data is structured. We experiment with two variants of the model, one that works with fine-grained POS tags and one that uses coarse-grained tags.

Basically, the difference between fine and coarse tags is that in fine tags we preserve and concatenate the POS representation of the affixes and clitics, while in coarse tags we eliminate affix representation and keep the POS for stems only. The distinction between fine and coarse tags is illustrated further with some examples in Table 2.

Word	Translit. / Gloss	Fine Tag	Coarse Tag
بيحبك	byHbk likes+you	prog_part +v+pron	Verb
هيعبرنا	hyEbrnA will+ consider+us	fut_part +v+pron	Verb
والعمر	wAlEmr and+ the+life	conj+det +noun	Noun
قلبك	qlbk your+heart	noun+pron	Noun
طايقك	TAyqk standing+you	adj+pron	Adj
عالأقل	EAl>ql at+ the+least	prep+det +adj	Adj

Table 2: Examples of unsegmented words with fine and coarse POS tags.

Our system achieves 92.38% accuracy with the coarse tags and 88.43% using the fine tags. The gap in performance is mostly due to the size of the tagset. The number fine POS tags observed in the data is 218, while there are only 28 coarse tags. It is to be mentioned that the reported accuracy for segmented words by Darwish et al. (2018) is 92.9%.

4.2 Features Used

Here we describe the features used in our deep learning model.

POS tags. We include POS tags, as predicted by the specially developed model described in 4.1 above, as a layer in the neural network model.

Word-level embeddings allow the learning algorithms to use large unlabeled data to generalize beyond the seen training data. We explore randomly initialized embeddings based on the seen training data and pre-trained embedding.

For pre-trained embedding, we use FastText (Bojanowski et al., 2017) on a corpus that we crawled from the web with a total size of 383,261,475 words, consisting of user-generated texts from Facebook posts (8,241,244), Twitter tweets (2,813,016), user comments on the news (95,241,480), and MSA texts of news articles (from Al-Jazeera and Al-Ahram) of 276,965,735 words. After building the embeddings, we run the

list of words in our dataset by the predictor in the word vector model to ensure that we get representations of all the words and reduce the number of OOVs (out of vocabulary words).

We find significant improvement using FastText embedding over the traditional word2vec representation (Mikolov et al., 2013). This is probably due to the utilization of sub-word (ex. prefixes or suffixes) information in the former.

Character-level CNNs. Although originally designed for image recognition, CNNs have proven effective for various NLP tasks due to their ability to encode character-level representations of words as well as extract sub-word information (Collobert et al., 2011; Chiu and Nichols, 2016; dos Santos and Guimarães, 2015).

Bi-LSTM Recurrent neural networks (RNN) are well suited for modeling sequential data, achieving ground-breaking results in many NLP tasks (e.g., machine translation). Bi-LSTMs (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) are capable of learning long-term dependencies and maintaining contextual features from both past and future states while avoiding the vanishing/exploding gradients problem. They consist of two separate bidirectional hidden layers that feed forward to the same output layer.

CRF is used jointly with bi-LSTMs to avoid the output label independence assumptions of bi-LSTMs and to impose sequence labeling constraints as in Lample et al. (2016). In our experiments with this task we find that CRF has a slight advantage over the softmax optimizer.

Brown clusters (BC). Brown clustering is an unsupervised learning method where words are grouped based on the contexts in which they appear (Brown et al., 1992). The assumption is that words that behave in similar ways tend to appear in similar contexts and hence belong to the same cluster. BCs can be learned from a large unlabeled corpus and have been shown to improve POS tagging as well as other sequence labelling tasks (Owoputi et al., 2013; Stratos and Collins, 2015). We test the effectiveness of using Brown clusters in the context of code-switching experimentation in a DNN model by training

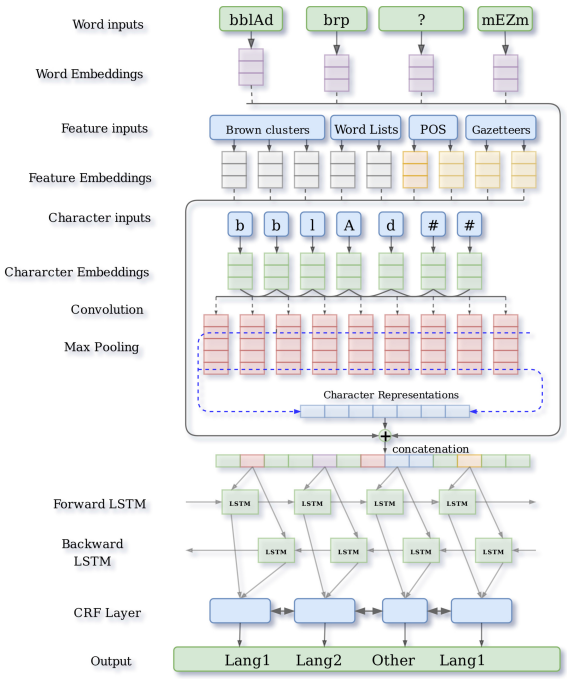

Figure 3: DNN Architecture.

BCs on our crawled code-switched corpus of 380 million words (mentioned above) with 100 Brown Clusters.

Named Entity Gazetteers We use a large collection of named entity gazetteers of 40,719 unique names from Attia et al. (2010), who collected named entities from the Arabic Wikipedia, and Benajiba et al. (2007), who annotated a corpus as part of a named entity recognition system. The assumption is that the gazetteer will enhance the system's recognition of NE's which constitutes between 11 and 14% of the tags in the datasets. The feature is used as a binary class, i.e. whether the word is present in the gazetteer list or not.

Spell Checking Word List Dialectal lexicon and inflection can vary significantly from the standard one. Based on this assumption we check for each word whether or not it exists in a large word list of fully inflected MSA words (Attia et al., 2012). The word list contains 9,196,215 and is obtained from the web as an open source resource [1].

The architecture of our model (with the best performance) is shown in Figure 3. For each word in the sequence, the CNN computes the character-level representation with character embeddings as inputs. Then the character-level rep-

[1] https://sourceforge.net/projects/arabic-wordlist/

resentation vector is concatenated with both word embeddings vector and feature embedding vectors (Brown Clusters, POS, and Gazetteers) to feed into the bi-LSTM layer. Finally, an affine transformation followed by a CRF is applied over the hidden representation of the bi-LSTM to obtain the probability distribution over all the code-switching labels. Training is performed using stochastic gradient descent with a momentum of 0.9 and batch size equal to 150. We employ dropout (Hinton et al., 2012) to mitigate over-fitting, and early-stopping (Caruana et al., 2000) (with patience of 35). We further use the hyper-parameters detailed in Table 3.

Layer	Hyper-Parameters	Value
Word Emb.	dimension	300
Characters Emb.	dimension	100
Characters CNN	window size	4
	number of filters	40
POS Emb.	dimension	166
Clustering Emb.	dimension	100
Gazetteer Emb.	dimension	2
Bi-LSTM	state size	100
Dropout	dropout rate	0.5
	batch size	150

Table 3: Parameter fine-tuning

5 Experiments and Results

We conduct a number experiments with different layers in the neural network model stacked on top of each other, making use of word and character representation, POS, FastText pre-trained embeddings, and other features. This allows us to see the significance of each feature and how it contributes to the overall performance of the system. The experiments are shown in Table 4.

The results in Table 4 are reported for the f-score measure on the validation set, except for the last row which gives the best model results on the test set. The results generally show that the DNN model is incrementally improving by adding more features and external resources. The best result is obtained with the aggregation of all features, excluding the SP (spell checking word list).

In the training data, $lang1$ (MSA) is the majority class representing 68.7% of the labels. We use majority voting as the baseline in order to detect if

#	Experiments	f-score	averaged f-score
1	Baseline (majority voting)	30.97	7.88
2	POS-coarse	66.19	40.57
3	POS-fine	72.99	45.28
4	Words	83.78	55.78
5	Words+POS-fine	84.68	57.06
6	Chars	84.02	56.52
7	Words+Chars	84.87	57.36
8	Words+Chars +POS-fine	86.47	58.15
9	Words+Chars +POS+BC	89.18	59.71
10	Words+Chars +POS+BC+GZ	89.21	59.63
11	Words+Chars +POS+BC+GZ +Embed	91.90	61.33
12	Words+Chars +POS+Embed +BC+GZ+SP	91.48	61.02
13	Words+Chars +POS+BC+GZ +Embed+PP	**91.92**	**61.35**
	Results on Test set	88.92	50.48

Table 4: DNN experiments and Results. Abbreviations: BC: Brown Clusters, GZ: named entity gazetteer, SP: Spelling word list, PP: post-processing

POS tags alone do send any positive signal to the model at all. We note that the baseline is very low which is due to the fact that the tag distribution in the training set is disproportionate with both the validation and the test set, where $lang1$ represents only 30.96% and 28.10% of the data respectively.

It is to be noted that we apply post-processing (PP) to the output of the prediction. The idea is that foreign words (words written in Latin script), punctuation marks, user names (words starting with the '@' sign), and hashtags (words starting with the '#' sign) should all be assigned the $other$ tag. As these are deterministic cases, we develop a post-process procedure to correct errors in the predictions of the probabilistic model, and to make sure that they are assigned the right tag.

Our experiments show that POS tags do give a strong signal to the network that leads to a significant improvement over the baseline, from 30.97% to 66.19% using coarse-grained POS features and 72.99% using the fine-grained tags. We also no-

Labels	Token Count	Token Ratio %	Samih et al. (2016)	Current System
ambiguous	10	0.05	0.00	0.00
lang1	6,406	30.96	0.88	0.91
lang2	9,355	45.22	0.92	0.93
mixed	2	0.01	0.00	0.00
ne	3,024	14.62	0.84	0.86
other	1,891	9.14	0.97	0.98
Accuracy	–	–	0.900	0.919

Table 5: F1 score token level comparison between Samih et al. (2016) and the current system on the development dataset.

Labels	Token Count	Token Ratio %	Samih et al. (2016)	Current System
ambiguous	117	0.57	0.000	0.00
unk	26	0.13	0.000	0.00
lang1	5,804	28.10	0.854	0.860
lang2	9,630	46.62	0.904	0.913
mixed	1	0.00	0.000	0.000
ne	2,363	11.31	0.777	0.789
other	2,743	13.28	0.957	0.965
Accuracy	–	–	0.879	0.889

Table 6: F1 score token level comparison between Samih et al. (2016) and the current system on the test dataset.

tice that using the predicted fine-grained POS is significantly more helpful than using the predicted coarse-grained one (although the prediction accuracy for fine-grained tags is lower). This is probably because the fine-grained POS tags encode more lexical information (related to clitics and affixes) that can have distinctive combinations. Adel et al. (2015) claimed that part-of-speech (POS) tags can predict CS points more reliably than words themselves, but our results show that words still give a stronger signal than POS tags alone.

We also notice that Brown Clusters, named entity gazetteers and FastText pre-trained embeddings contribute to incrementally improve the performance of the system. Unfortunately adding information from the spelling word list did not show any improvement on the system, and this is why it is removed from the final system architecture.

Now we compare our best model to the state-of-the-art system of Samih et al. (2016), which won the 2016 Second Shared Task on Language Identification in Code-Switched Data (Molina et al., 2016) on the MSA–EA dataset. We compare the performance of the two systems in terms of f-score accuracy on both the development and test set, in Table 5 and Table 6 respectively. We also include the number of instances and the ratio percentage for each label. As the tables show, the category $lang2$ constitutes the majority class for both

	amb	ne	mixed	other	L1	L2
amb	0	0	0	0	1	9
ne	0	2507	0	14	277	226
mx	0	0	0	0	0	2
other	0	4	0	1844	7	36
L1	12	121	0	9	5931	333
L2	1	188	0	9	423	8734

Table 7: Confusion matrix for the development dataset.

the validation and test sets (45.22% and 46.62% respectively), contrary to the training set where $lang1$ makes up 68.70% of the labels.

For the development set our system outperforms that of Samih et al. (2016) by 1.9% absolute with significant gains for $lang1$ (3% absolute) and ne (2% absolute). For the test set our system again outperforms that of Samih et al. (2016) by 1.0% absolute with the gain spread almost evenly across all labels.

Table 7 presents the confusion matrix for the validation set, which shows that ne suffers the largest confusion as it gets mixed up as either $lang2$ (EA) or $lang1$ (MSA). This is due to the fact that many named entities in Arabic can also be used as ordinary words, and, unlike English, there is no case marking or other orthographic features that can superficially distinguish the two. For example, the word كريم krym, can mean either "Kareem" as an ne or "generous" as an adjective, and جمال jmAl can mean "Jamal" as an ne or "beauty" as a noun. The second largest confusion is between $lang1$ and $lang2$, where we find that a considerable amount of the mix-up coming from function words, such as و wa "and", أو >aw "or" and إلى <ilY "to", which can equally be used as either $lang1$ or $lang2$, depending on the context.

6 Conclusion

We have presented a neural network system for conducting word-level code-switching identification. Our system outperforms the current state-of-the-art, and we show that adding linguistic features can contribute to improving the performance of the deep learning models. We show that POS tagging gives a strong positive signal for code-switching prediction. We also examine the syntactic patterns in diglossic code-switching, and observe that dialects show a bias in the choice of word categories toward dialectal function words over content words.

References

Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. Syntactic and semantic features for code-switching factored language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):431–440.

Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. 2013a. Recurrent neural network language modeling for code switching conversational speech. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8411–8415. IEEE.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. 2013b. Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 206–211.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor Soto, and Julia Hirschberg. 2016. Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107.

Mohammed Attia, Pavel Pecina, Younes Samih, Khaled Shaalan, and Josef Van Genabith. 2012. Improved spelling error detection and correction for arabic. *Proceedings of COLING 2012: Posters*, pages 103–112.

Mohammed Attia, Antonio Toral, Lamia Tounsi, Monica Monachini, and Josef van Genabith. 2010. An automatically built named entity lexicon for Arabic. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Languages Resources Association (ELRA).

Peter Auer. 1999. From codeswitching via language mixing to fused lects: Toward a dynamic typology of bilingual speech. *International journal of bilingualism*, 3(4):309–332.

George Carpenter Barker. 1972. *Social Functions of Language in a Mexican-American Community*, volume 22. University of Arizona Press.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 143–153. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Eyamba G Bokamba. 1989. Are there syntactic constraints on code-mixing? *World Englishes*, 8(3):277–292.

Erman Boztepe. 2003. Issues in code-switching: Competing theories and models. *Columbia University Working Papers in TESOL and Applied Linguistics*.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Rich Caruana, Steve Lawrence, and Lee Giles. 2000. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*, pages 402–408.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11, Austin, TX.

Kunal Chakma and Amitava Das. 2016. Cmir: A corpus for evaluation of code mixed information retrieval of hindi-english tweets. *Computación y Sistemas*, 20(3):425–434.

Hok Shing Chan. 2009. Code-switching between typologically distinct languages. *The Cambridge handbook of linguistic code-switching*, pages 182–198.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. Multi-dialect arabic pos tagging: A crf approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, pages 93–98, Miyazaki, Japan.

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*, pages 1–12.

Penelope Gardner-Chloros. 1991. *Language selection and switching in Strasbourg*. Oxford University Press.

François Grosjean. 1989. Neurolinguists, beware! the bilingual is not two monolinguals in one person. *Brain and language*, 36(1):3–15.

Nizar Habash, Owen Rambow, Mona Diab, and Reem Kanjawi-Faraj. 2008. Guidelines for annotation of arabic dialectness. In *Proceedings of the LREC*

Workshop on HLT & NLP within the Arabic world, pages 49–53.

Monica Heller. 2010. *Codeswitching: Anthropological and Sociolinguistic Perspectives*, volume 48. Walter de Gruyter.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Istvan Kecskes. 2006. The dual language model to explain code-switching: A cognitive-pragmatic approach. *Intercultural Pragmatics*, 3(3):257–283.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

John M Lipski. 2005. Code-switching or borrowing? no sé so no puedo decir, you know. In *Selected proceedings of the second workshop on Spanish sociolinguistics*, pages 1–15. Cascadilla Proceedings Project Somerville, MA.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

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 Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Pieter Muysken. 1995. *Code-switching and grammatical theory*, page 177–198. Cambridge University Press.

Pieter Muysken, Carmen Pena Díaz, Pieter Cornelis Muysken, et al. 2000. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Chad Nilep. 2006. "code switching" in sociocultural linguistics. *Colorado Research in Linguistics*, 19(1):1.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390. Association for Computational Linguistics.

David J Parkin. 1974. Language switching in nairobi. *Language in Kenya*, pages 189–216.

Anat Prior and Tamar H Gollan. 2011. Good language-switchers are good task-switchers: Evidence from spanish–english and mandarin–english bilinguals. *Journal of the International Neuropsychological Society*, 17(4):682–691.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. Answer ka type kya he?: Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio. Association for Computational Linguistics.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59.

Cicero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China. Association for Computational Linguistics.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Mamokgethi Setati. 1998. Code-switching in a senior primary class of second-language mathematics learners. *For the Learning of Mathematics*, 18(1):34–40.

Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 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*, pages 62–72.

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 973–981. Association for Computational Linguistics.

Thamar Solorio and Yang Liu. 2008b. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060. Association for Computational Linguistics.

Karl Stratos and Michael Collins. 2015. Simple semi-supervised pos tagging. In *VS@ HLT-NAACL*, pages 79–87.

Peter Trudgill. 1986. *Dialects in contact*. B. Blackwell.

Stavroula Tsiplakou. 2009. Code-switching and code-mixing between related varieties: establishing the blueprint. *The International Journal of Humanities*, 6(12):49–66.

David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. 2015. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–8.

Donald D Walsh. 1969. Bilingualism and bilingual education: a guest editorial. *Foreign Language Annals*, 2(3):298–303.

Michael Yoder, Shruti Rijhwani, Carolyn Rosé, and Lori Levin. 2017. Code-switching as a social act: The case of arabic wikipedia talk pages. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 73–82.