Sentence Level Dialect Identification in Arabic

Heba Elfardy
Department of Computer Science
Columbia University
heba@cs.columbia.edu

Mona Diab
Department of Computer Science
The George Washington University
mtdiab@gwu.edu

Abstract

This paper introduces a supervised approach for performing sentence level dialect identification between Modern Standard Arabic and Egyptian Dialectal Arabic. We use token level labels to derive sentence-level features. These features are then used with other core and meta features to train a generative classifier that predicts the correct label for each sentence in the given input text. The system achieves an accuracy of 85.5% on an Arabic online-commentary dataset outperforming a previously proposed approach achieving 80.9% and reflecting a significant gain over a majority baseline of 51.9% and two strong baseline systems of 78.5% and 80.4%, respectively.

1 Introduction

The Arabic language exists in a state of Diglossia (Ferguson, 1959) where the standard form of the language, Modern Standard Arabic (MSA) and the regional dialects (DA) live side-by-side and are closely related. MSA is the language used in education, scripted speech and official settings while DA is the native tongue of Arabic speakers. Arabic dialects may be divided into five main groups: Egyptian (including Libyan and Sudanese), Levantine (including Lebanese, Syrian, Palestinian and Jordanian), Gulf, Iraqi and Moroccan (Maghrebi) (Habash, 2010). Even though these dialects did not originally exist in a written form, they are pervasively present in social media text (normally mixed with MSA) nowadays. DA does not have a standard orthography leading to many spelling variations and inconsistencies. Linguistic Code switching (LCS) between MSA and DA happens both intra-sententially and inter-sententially. LCS in Arabic poses a serious challenge for almost all NLP tasks since MSA and DA differ on all levels of linguistic representation. For example, MSA trained tools perform very badly when applied directly to DA or to a code-switched DA-MSA text. Hence a need for a robust dialect identification tool as a preprocessing step arises both on the word and sentence levels.

In this paper, we focus on the problem of dialect identification on the sentence level. We propose a supervised approach for identifying whether a given sentence is prevalently MSA or Egyptian DA (EDA). The system uses the approach that was presented in (Elfardy et al., 2013) to perform token dialect identification. The token level decisions are then combined with other features to train a generative classifier that tries to predict the class of the given sentence. The presented system outperforms the approach presented by Zaidan and Callison-Burch (2011) on the same dataset using 10-fold cross validation.

2 Related Work

Dialect Identification in Arabic is crucial for almost all NLP tasks, yet most of the research in Arabic NLP, with few exceptions, is targeted towards MSA. Biadsy et al. (2009) present a system that identifies dialectal words in speech and their dialect of origin through the acoustic signals. Salloum and Habash (2011) tackle the problem of DA to English Machine Translation (MT) by pivoting through MSA. The authors present a system that applies transfer rules from DA to MSA then uses state of the art MSA to English MT system. Habash et al. (2012) present CODA, a Conventional Orthography for Dialectal Arabic that aims to standardize the orthography of all the variants of DA while Dasigi and Diab (2011) present an unsupervised clustering approach to identify orthographic variants in DA. Zaidan and Callison-Burch (2011) crawl a large dataset of MSA-DA news' commentaries. The authors annotate part of the dataset for sentence-level dialectalness on

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 456–461,
Sofia, Bulgaria, August 4-9 2013.

Amazon Mechanical Turk and try a language modeling (LM) approach to solve the problem. In Elfardy and Diab (2012a), we present a set of guidelines for token-level identification of dialectalness while in (Elfardy and Diab, 2012b), (Elfardy et al., 2013) we tackle the problem of token-level dialect-identification by casting it as a code-switching problem.

3 Approach to Sentence-Level Dialect Identification

We present a supervised system that uses a Naive Bayes classifier trained on gold labeled data with sentence level binary decisions of either being MSA or DA.

3.1 Features

The proposed supervised system uses two kinds of features: (1) Core Features, and (2) Meta Features.

3.1.1 Core Features:

These features indicate how dialectal (or non dialectal) a given sentence is. They are further divided into: (a) Token-based features and (b) Perplexity-based features.

3.1.1.1 Token-based Features: We use the approach that was presented in (Elfardy et al., 2013) to decide upon the class of each word in the given sentence. The aforementioned approach relies on language models (LM) and MSA and EDA Morphological Analyzer to decide whether each word is (a) MSA, (b) EDA, (c) Both (MSA & EDA) or (d) OOV. We use the token-level class labels to estimate the percentage of EDA words and the percentage of OOVs for each sentence. These percentages are then used as features for the proposed model. The following variants of the underlying token-level system are built to assess the effect of varying the level of preprocessing on the underlying LM on the performance of the overall sentence level dialect identification process: (1) Surface, (2) Tokenized, (3) CODAfied, and (4) Tokenized-CODA. We use the following sentence to show the different techniques:

كده حرام وكتير علينا *kdh HrAm wktyr ElynA*

1. **Surface LMs:** No significant preprocessing is applied apart from the regular initial clean up of the text which includes removal of URLs, normalization of speech effects such as reducing all redundant letters in a word to a standardized form, eg. the elongated form of the word كتير **ktyr**[1] 'a lot' which could be rendered in the text as كتتتتييييير *ktttyyyyr* is reduced to كتتتييير *ktttyyyr* (specifically three repeated letters instead of an unpredictable number of repetitions, to maintain the signal that there is a speech effect which could be a DA indicator).
   ex. كده حرام وكتير علينا
   kdh HrAm wktyr ElynA

2. **Orthography Normalized (CODAfied) LM:** since DA is not originally a written form of Arabic, no standard orthography exists for it. Habash et al. (2012) attempt to solve this problem by presenting CODA, a conventional orthography for writing DA. We use the implementation of CODA presented in CODAfy (Eskander et al., 2013), to build an orthography-normalized LM. While CODA and its applied version using CODAfy solve the spelling inconsistency problem in DA, special care must be taken when using it for our task since it removes valuable dialectalness cues. For example, the letter ث (v in Buckwalter (BW) Transliteration) is converted into the letter ت (t in BW) in a DA context. CODA suggests that such cases get mapped to the original MSA phonological variant which might make the dialect identification problem more challenging. On the other hand, CODA solves the sparseness issue by mapping multiple spelling-variants to the same orthographic form leading to a more robust LM.
   ex. كده حرام وكثير علينا
   kdh HrAm wkvyr ElynA

3. **Tokenized LM:** D3 tokenization-scheme is applied to all data using MADA (Habash et al., 2009) (an MSA Tokenizer) for the MSA corpora, and MADA-ARZ (Habash et al., 2013) (an EDA tokenizer) for the EDA corpora. For building the tokenized LM, we maintain clitics and lexemes. Some clitics are unique to MSA while others are unique to EDA so maintaining them in the LM is helpful, eg. the negation enclitic ش *$* is only used in EDA but it could be seen with an MSA/EDA homograph, maintaining the enclitic in the LM facilitates the identification

[1] We use Buckwalter transliteration scheme http://www.qamus.org/transliteration.htm

of the sequence as being EDA. 5-grams are used for building the tokenized LMs (as opposed to 3-grams for the surface LMs)
ex. كده حرام و كتير علي نا
kdh HrAm w+ ktyr Ely +nA

4. **Tokenized & Orthography Normalized LMs: (Tokenized-CODA)** The data is tokenized as in (3) then orthography normalization is applied to the tokenized data.

ex. كده حرام و كثير علي نا
kdh HrAm w+ kvyr Ely +nA

In addition to the underlying token-level system, we use the following token-level features:

1. Percentage of words in the sentence that is analyzable by an MSA morphological analyzer.
2. Percentage of words in the sentence that is analyzable by an EDA morphological analyzer.
3. Percentage of words in the sentence that exists in a precompiled EDA lexicon.

3.1.1.2 Perplexity-based Features: We run each sentence through each of the MSA and EDA LMs and record the perplexity for each of them. The perplexity of a language model on a given test sentence; $S(w_1, .., w_n)$ is defined as:

$$perplexity = (2)^{-(1/N)\sum_i \log_2(p(w_i|h_i))} \quad (1)$$

where N is the number of tokens in the sentence and h_i is the history of token w_i.

The perplexity conveys how confused the LM is about the given sentence so the higher the perplexity value, the less probable that the given sentence matches the LM.[2]

3.1.2 Meta Features.

These are the features that do not directly relate to the dialectalness of words in the given sentence but rather estimate how informal the sentence is and include:

- The percentage of punctuation, numbers, special-characters and words written in Roman script.
- The percentage of words having word-lengthening effects.
- Number of words & average word-length.
- Whether the sentence has consecutive repeated punctuation or not. (Binary feature, yes/no)
- Whether the sentence has an exclamation mark or not. (Binary feature, yes/no)
- Whether the sentence has emoticons or not. (Binary feature, yes/no)

[2]We repeat this step for each of the preprocessing schemes explained in section 3.1.1.1

3.2 Model Training

We use the WEKA toolkit (Hall et al., 2009) and the derived features to train a Naive-Bayes classifier. The classifier is trained and cross-validated on the gold-training data for each of our different configurations (Surface, CODAfied, Tokenized & Tokenized-CODA).

We conduct two sets of experiments. In the first one, Experiment Set A, we split the data into a training set and a held-out test set. In the second set, Experiment Set B, we use the whole dataset for training without further splitting. For both sets of experiments, we apply 10-fold cross validation on the training data. While using a held-out test-set for evaluation (in the first set of experiments) is a better indicator of how well our approach performs on unseen data, only the results from the second set of experiments are directly comparable to those produced by Zaidan and Callison-Burch (2011).

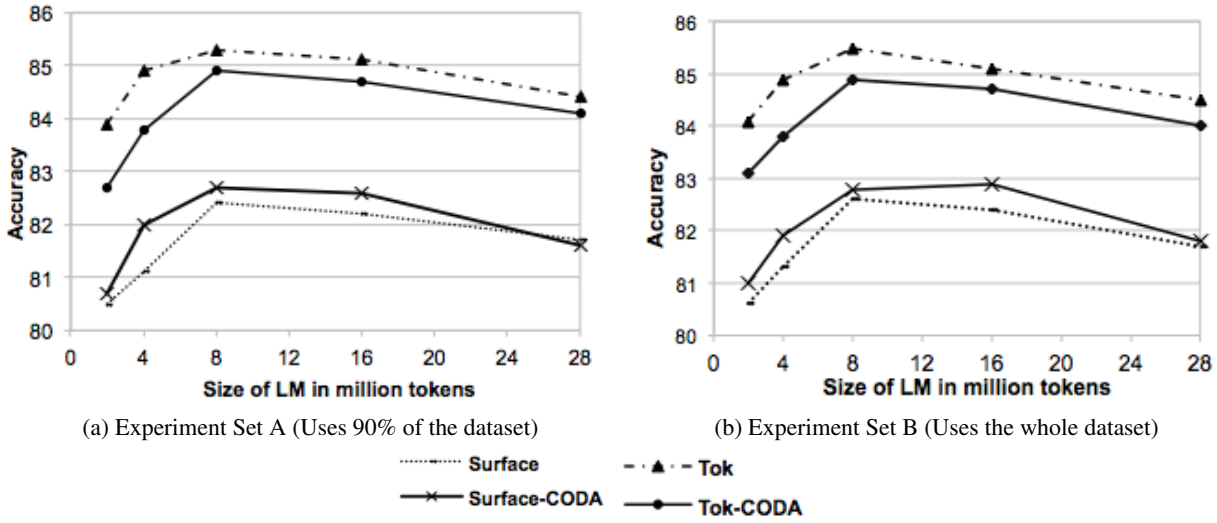
4 Experiments

4.1 Data

We use the code-switched EDA-MSA portion of the crowd source annotated dataset by Zaidan and Callison-Burch (2011). The dataset consists of user commentaries on Egyptian news articles. Table 1 shows the statistics of the data.

	MSA Sent.	EDA Sent.	MSA Tok.	EDA Tok.
Train	12,160	11,274	300,181	292,109
Test	1,352	1,253	32,048	32,648

Table 1: Number of EDA and MSA sentences and tokens in the training and test datasets. In Experiment Set A only the train-set is used to perform a 10-fold cross-validation and the test-set is used for evaluation. In experiment Set B, all data is used to perform the 10-fold cross-validation.



(a) Experiment Set A (Uses 90% of the dataset)

(b) Experiment Set B (Uses the whole dataset)

Figure 1: Learning curves for the different configurations (obtained by applying 10-fold cross validation on the training set.)

4.2 Baselines

We use four baselines. The first of which is a majority baseline (Maj-BL); that assigns all the sentences the label of the most frequent class observed in the training data. The second baseline (Token-BL) assumes that the sentence is EDA if more than 45% of its tokens are dialectal otherwise it assumes it is MSA.[3] The third baseline (Ppl-BL) runs each sentence through MSA & EDA LMs and assigns the sentence the class of the LM yielding the lower perplexity value. The last baseline (OZ-CCB-BL) is the result obtained by Zaidan and Callison-Burch (2011) which uses the same approach of our third baseline, Ppl-BL.[4] For Token-BL and Ppl-BL, the performance is calculated for all LM-sizes of the four different configurations: Surface, CODAfied, Tokenized, Tokenized-CODA and the best performing configuration on the cross-validation set is used as the baseline system.

4.3 Results & Discussion

For each of the different configurations, we build a learning curve by varying the size of the LMs between 2M, 4M, 8M, 16M and 28M tokens. Figures 1a and 1b show the learning curves of the different configurations on the cross-validation set for experiments A & B respectively. In Table 2 we note that both CODA and Tokenized solve the data-sparseness issue hence they produce better results than Surface experimental condition. However, as mentioned earlier, CODA removes some dialectalness cues so the improvement resulting from using CODA is much less than that from using tokenization. Also when combining CODA with tokenization as in the condition Tokenized-CODA, the performance drops since in this case the sparseness issue has been already resolved by tokenization so adding CODA only removes dialectalness cues. For example وكتير *wktyr* 'and a lot' does not occur frequently in the data so when performing the tokenization it becomes و كتير *w+ ktyr* which on the contrary is frequent in the data. Adding

Condition	Exp. Set A	Exp. Set B
Maj-BL	51.9	51.9
Token-BL	79.1	78.5
Ppl-BL	80.4	80.4
OZ-CCB-BL	N/A	80.9
Surface	82.4	82.6
CODA	82.7	82.8
Tokenized	**85.3**	**85.5**
Tokenized-CODA	84.9	84.9

Table 2: Performance Accuracies of the different configurations of the 8M LM (best-performing LM size) using 10-fold cross validation against the different baselines.

[3] We experimented with different thresholds (15%, 30%, 45%, 60% and 75%) and the 45% threshold setting yielded the best performance

[4] This baseline can only be compared to the results of the second set of experiments.

Condition	Test Set
Maj-BL	51.9
Token-BL	77
Ppl-BL	81.1
Tokenized	**83.3**

Table 3: Performance Accuracies of the best-performing configuration (Tokenized) on the held-out test set against the baselines Maj-BL, Token-BL and Ppl-BL.

Orthography-Normalization converts it to و كثير *w+ kvyr* which is more MSA-like hence the confusability increases.

All configurations outperform all baselines with the Tokenized configuration producing the best results. The performance of all systems drop as the size of the LM increases beyond 16M tokens. As indicated in (Elfardy et al., 2013) as the size of the MSA & EDA LMs increases, the shared ngrams increase leading to higher confusability between the classes of tokens in a given sentence. Table 3 presents the results on the held out dataset compared against three of the baselines, Maj-BL, Token-BL and Ppl-BL. We note that the Tokenized condition, the best performing condition, outperforms all baselines with a significant margin.

5 Conclusion

We presented a supervised approach for sentence level dialect identification in Arabic. The approach uses features from an underlying system for token-level identification of Egyptian Dialectal Arabic in addition to other core and meta features to decide whether a given sentence is MSA or EDA. We studied the impact of two types of preprocessing techniques (Tokenization and Orthography Normalization) as well as varying the size of the LM on the performance of our approach. The presented approach produced significantly better results than a previous approach in addition to beating the majority baseline and two other strong baselines.

Acknowledgments

This work is supported by the Defense Advanced Research Projects Agency (DARPA) BOLT program under contract number HR0011-12-C-0014.

References

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at the meeting of the European Association for Computational Linguistics (EACL), Athens, Greece.*

Pradeep Dasigi and Mona Diab. 2011. Codact: Towards identifying orthographic variants in dialectal arabic. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (ICJNLP), Chiangmai, Thailand.*

Heba Elfardy and Mona Diab. 2012a. Simplified guidelines for the creation of large scale dialectal arabic annotations. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey.*

Heba Elfardy and Mona Diab. 2012b. Token level identification of linguistic code switching. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING),Mumbai, India.*

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2013. Code Switch Point Detection in Arabic. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB2013)*, MediaCity, UK, June.

Ramy Eskander, Nizar Habash, Owen Rambow, and Nadi Tomeh. 2013. Processing Spontaneous Orthography. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA.

Ferguson. 1959. *Diglossia. Word 15. 325340.*

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt*, pages 102–109.

Nizar Habash, Mona Diab, and Owen Rabmow. 2012. Conventional orthography for dialectal arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Istanbul.*

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA.

Nizar Habash. 2010. Introduction to arabic natural language processing. *Advances in neural information processing systems.*

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of ACL*, pages 37–41.