AIDA2: A Hybrid Approach for Token and Sentence Level Dialect Identification in Arabic

Mohamed Al-Badrashiny, Heba Elfardy[†] and Mona Diab
Department of Computer Science, The George Washington University
{badrashiny,mtdiab}@gwu.edu
[†]Department of Computer Science, Columbia University
[†]heba@cs.columbia.edu

Abstract

In this paper, we present a hybrid approach for performing token and sentence levels Dialect Identification in Arabic. Specifically we try to identify whether each token in a given sentence belongs to Modern Standard Arabic (MSA), Egyptian Dialectal Arabic (EDA) or some other class and whether the whole sentence is mostly EDA or MSA. The token level component relies on a Conditional Random Field (CRF) classifier that uses decisions from several underlying components such as language models, a named entity recognizer and and a morphological analyzer to label each word in the sentence. The sentence level component uses a classifier ensemble system that relies on two independent underlying classifiers that model different aspects of the language. Using a feature-selection heuristic, we select the best set of features for each of these two classifiers. We then train another classifier that uses the class labels and the confidence scores generated by each of the two underlying classifiers to decide upon the final class for each sentence. The token level component yields a new state of the art F-score of 90.6% (compared to previous state of the art of 86.8%) and the sentence level component yields an accuracy of 90.8% (compared to 86.6% obtained by the best state of the art system).

1 Introduction

In this age of social media ubiquity, we note the pervasive presence of informal language mixed in with formal language. Degree of mixing formal and informal language registers varies across languages making it ever harder to process. The problem is quite pronounced in Arabic where the difference between the formal modern standard Arabic (MSA) and the informal dialects of Arabic (DA) could add up to a difference in language morphologically, lexically, syntactically, semantically and pragmatically, exacerbating the challenges for almost all NLP tasks. MSA is used in formal settings, edited media, and education. On the other hand the spoken, and, currently written in social media and penetrating formal media, are the informal vernaculars. There are multiple dialects corresponding to different parts of the Arab world: (1) Egyptian, (2) Levantine, (3) Gulf, (4) Moroccan, and, (5) Iraqi. For each one of these sub-dialectal variants exist. Speakers/writers code switch between the two forms of the language especially in social media text both inter and intra sententially. Automatically identifying code-switching between variants of the same language (Dialect Identification) is quite challenging due to the lexical overlap and significant semantic and pragmatic variation yet it is crucial as a preprocessing step before building any Arabic NLP tool. MSA trained tools perform very badly when applied directly to DA or to intrasentential code-switched DA and MSA text (ex. *Alfryq fAz bAlEAfyp bs tSdr qA}mp Aldwry*, where the words correspond to MSA MSA DA DA MSA MSA MSA, respectively)[1]. Dialect Identification has been shown to be an important preprocessing step for statistical machine Translation (SMT). (Salloum et al., 2014) explored the impact of using Dialect Identification on the performance of MT and found that it improves the results. They trained four different SMT systems; (a) DA-to-English SMT, (b) MSA-to-English SMT, (c) DA + MSA-to-English SMT, and (d) DA-to-English hybrid MT system and treated the task of choosing

[1]We use Buckwalter transliteration scheme to represent Arabic in Romanized script throughout the paper. http://www.qamus.org/transliteration.htm

Proceedings of the 19th Conference on Computational Language Learning, pages 42–51,
Beijing, China, July 30-31, 2015.

which SMT system to invoke as a classification task. They built a classifier that uses various features derived from the input sentence and that indicate, among other things, how dialectal the input sentence is and found that this approach improved the performance by 0.9% BLEU points.

In this paper, we address the problem of token and sentence levels dialect identification in Arabic, specifically between Egyptian Arabic and MSA. For the token level task, we treat the problem as a sequence labeling task by training a CRF classifier that relies on the decisions made by a language model, a morphological analyzer, a shallow named entity recognition system, a modality lexicon and other features pertaining to the sentence statistics to decide upon the class of each token in the given sentence. For the sentence level task we resort to a classifier ensemble approach that combines independent decisions made by two classifiers and use their decisions to train a new one. The proposed approaches for both tasks significantly beat the current state of the art performance with a significant margin, while creating a pipelined system.

2 Related Work

Dialect Identification in Arabic has recently gained interest among Arabic NLP researchers. Early work on the topic focused on speech data. Biadsy et al. (2009) presented a system that identifies dialectal words in speech through acoustic signals. More recent work targets textual data. The main task for textual data is to decide the class of each word in a given sentence; whether it is *MSA*, *EDA* or some other class such as Named-Entity or punctuation and whether the whole sentence is mostly *MSA* or *EDA*. The first task is referred to as "Token Level Dialect Identification" while the second is "Sentence Level Dialect Identification".

For sentence level dialect identification in Arabic, the most recent works are (Zaidan and Callison-Burch, 2011), (Elfardy and Diab, 2013), and (Cotterell and Callison-Burch, 2014a). Zaidan and Callison-Burch (2011) annotate MSA-DA news commentaries on Amazon Mechanical Turk and explore the use of a language-modeling based approach to perform sentence-level dialect identification. They target three Arabic dialects; Egyptian, Levantine and Gulf and develop different models to distinguish each of them against the others and against MSA. They achieve an accuracy of 80.9%, 79.6%, and 75.1% for the Egyptian-MSA, Levantine-MSA, and Gulf-MSA classification, respectively. These results support the common assumption that Egyptian, relative to the other Arabic dialectal variants, is the most distinct dialect variant of Arabic from MSA. Elfardy and Diab (2013) propose a supervised system to perform Egyptian Arabic Sentence Identification. They evaluate their approach on the Egyptian part of the dataset presented by Zaidan and Callison-Burch (2011) and achieve an accuracy of 85.3%. Cotterell and Callison-Burch (2014b) extend Zaidan and Callison-Burch (2011) work by handling two more dialects (Iraqi and Moroccan) and targeting a new genre, specifically tweets. Their system outperforms Zaidan and Callison-Burch (2011) and Elfardy and Diab (2013), achieving a classification accuracy of 89%, 79%, and 88% on the same Egyptian, Levantine and Gulf datasets. For token level dialect identification, King et al. (2014) use a language-independent approach that 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[2] and ldig.[3] 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) use character n-grams to train a maximum entropy classifier that identifies whether a word is *MSA* or *EDA*. 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.

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

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

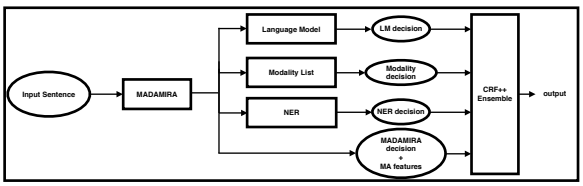

Figure 1: Token-level identification pipeline

In our previously published system *AIDA* (Elfardy et al., 2014) we use a weakly supervised rule based approach that relies on a language model to tag each word in the given sentence to be *MSA*, *EDA*, or *unk*. We then use the LM decision for each word in the given sentence/tweet and combine it with other morphological information, in addition to a named entity gazetteer to decide upon the final class of each word.

3 Approach

We introduce *AIDA*2. This is an improved version of our previously published tool *AIDA* (Elfardy et al., 2014). It tackles the problems of dialect identification in Arabic both on the token and sentence levels in mixed modern standard Arabic *MSA* and Egyptian dialect *EDA* text. We first classify each word in the input sentence to be one of the following six tags as defined in the shared task for "Language Identification in Code-Switched Data" in the first workshop on computational approaches to code-switching [ShTk](Solorio et al., 2014):

- *lang1*: If the token is *MSA* (ex. AlwAqE, "The reality")
- *lang2*: If the token is *EDA* (ex. m$, "Not")
- *ne*: If the token is a named entity (ex. >mrykA, "America")
- *ambig*: If the given context is not sufficient to identify the token as *MSA* or *EDA* (ex. slAm Elykm, "Peace be upon you")
- *mixed*: If the token is of mixed morphology (ex. b>myT meaning "I'm always removing")
- *other*: If the token is or is attached to any non Arabic token (ex. numbers, punctuation, Latin character, emoticons, etc)

The fully tagged tokens in the given sentence are then used in addition to some other features to classify the sentence as being mostly *MSA* or *EDA*.

3.1 Token Level Identification

Identifying the class of a token in a given sentence requires knowledge of its surrounding tokens since these surrounding tokens can be the trigger for identifying a word as being *MSA* or *EDA*. This suggests that the best way to approach the problem is by treating it as a sequence labeling task. Hence we use a Conditional Random Field (CRF) classifier to classify each token in the input sentence. The CRF is trained using decisions from the following underlying components:

- *MADAMIRA*: is a publicly available tool for morphological analysis and disambiguation of *EDA* and *MSA* text (Pasha et al., 2014).[4] *MADAMIRA* uses *SAMA* (Maamouri et al., 2010) to analyze the *MSA* words and *CALIMA* (Habash et al., 2012) for the *EDA* words. We use *MADAMIRA* to tokenize both the language model and input sentences using D3 tokenization-scheme, the most detailed level of tokenization provided by the tool (ex. bAlfryq, "By the team" becomes "b+ Al+ fryq")(Habash and Sadat, 2006). This is important in order to maximize the Language Models (LM) coverage. Furthermore, we also use *MADAMIRA* to tag each token in the input sentence as *MSA* or *EDA* by tagging the source of the morphological analysis, if *MADAMIRA*. analyses the word using *SAMA*, then the token is tagged *MSA* while if the analysis comes from *CALIMA*, the token is tagged *EDA*. Out of vocabulary words are tagged *unk*.
- *Language Model*: is a D3-tokenized 5-grams language model. It is built using the 119K manually annotated words of the training data of the shared task ShTk in addition to 8M words from weblogs data (4M from *MSA* sources and 4M from *EDA* ones). The weblogs are automatically annotated based on their source, namely, if the source of the data is dialectal, all the words from this source are tagged as *EDA*. Otherwise they are tagged *MSA*. Since we are using a D3-tokenized data, all D3 tokens of a word are assigned the same tag of their corresponding word (ex. if the word "bAlfryq" is tagged *MSA*, then each of "b+", "Al+", and "fryq" is tagged *MSA*). During runtime, the *Language Model* classifier module creates a lattice of all possible tags for each word in the input sentence after it is being tokenized by *MADAMIRA*. Viterbi search algorithm (Forney, 1973) is then used to find the best sequence of tags for the given sentence. If the input sentence contains out of vocabulary

[4] http://nlp.ldeo.columbia.edu/madamira/

words, they are being tagged as *unk*. This module also provides a binary flag called "*isMixed*". It is "true" only if the LM decisions for the prefix, stem, and suffix are not the same.

- *Modality List*: ModLex (Al-Sabbagh et al., 2013) is a manually compiled lexicon of Arabic modality triggers (i.e. words and phrases that convey modality). It provides the lemma with a context and the class of this lemma (*MSA*, *EDA*, or both) in that context. In our approach, we match the lemma of the input word that is provided by *MADAMIRA* and its surrounding context with an entry in ModLex. Then we assign this word the corresponding class from the lexicon. If we find more than one match, we use the class of the longest matched context. If there is no match, the word takes *unk* tag. Ex. the word "Sdq" which means "told the truth" gets the class "both" in this context ">flH An Sdq" meaning "He will succeed if he told the truth".
- *NER*: this is a shallow named entity recognition module. It provides a binary flag "*isNE*" for each word in the input sentence. This flag is set to "true" if the input word has been tagged as *ne*. It uses a list of all sequences of words that are tagged as *ne* in the training data of ShTk in addition to the named-entities from ANERGazet (Benajiba et al., 2007) to identify the named-entities in the input sentence. This module also checks the POS provided by *MADAMIRA* for each input word. If a token is tagged as *noun_prop* POS, then the token is classified as *ne*.

Using these four components, we generate the following features for each word:.

- *MADAMIRA-features*: the input word, prefix, stem, suffix, POS, *MADAMIRA* decision, and associated confidence score;
- *LM-features*: the "*isMixed*" flag in addition to the prefix-class, stem-class, suffix-class and the confidence score for each of them as provided by the language model;
- *Modality-features*: the *Modality List* decision;
- *NER-features*: the "*isNE*" flag from the *NER*;
- *Meta-features*: "*isOther*" is a binary flag that is set to "true" only if the input word is a non Arabic token. And "*hasSpeechEff*" which is another binary flag set to "true" only if the input word has speech effects (i.e. word lengthening).

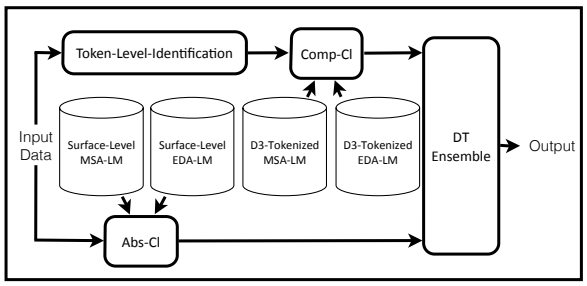

Figure 2: Sentence-level identification pipeline

We then use these features to train a CRF classifier using CRF++ toolkit (Sha and Pereira, 2003) and we set the window size to 16.[5] Figure 1 illustrates the different components of the token-level system.

3.2 Sentence Level Identification

For this level of identification, we rely on a classifier ensemble to generate the class label for each sentence. The underlying classifiers are trained on gold labeled data with sentence level binary decisions of either being *MSA* or *EDA*. Figure 2 shows the pipeline of the sentence level identification component. The pipeline consists of two main pathways with some pre-processing components. The first classifier (Comprehensive Classifier/*Comp-Cl*) is intended to cover dialectal statistics, token statistics, and writing style while the second one (Abstract Classifier/*Abs-Cl*) covers semantic and syntactic relations between words. The decisions from the two classifiers are fused together using a decision tree classifier to predict the final class of the input sentence.[6]

3.2.1 Comprehensive Classifier

The first classifier is intended to explicitly model detailed aspects of the language. We identify multiple features that are relevant to the task and we group them into different sets. Using the D3 tokenized version of the input data in addition to the classes provided by the "Token Level Identification" module for each word in the given sentence, we conduct a suite of experiments using the decision tree implementation by *WEKA* toolkit (Hall et al., 2009) to exhaustively search over all features in each group in the first phase, and then exhaustively search over all of the remaining features

[5]The window size is set empirically, we experimented with window sizes of 2, 4, 6, 8, 12.

[6]We experiment with different classifiers: Naive Bayes and Bayesian Network classifiers, but Decision Trees yielded the best results

from all groups to find the best combination of features that maximizes 10-fold cross-validation on the training data. We explore the same features used by Elfardy and Diab (2013) in addition to three other features that we refer to as "Modality Features". The full list of features include:

- *Perplexity-Features [PF]*: We run the tokenized input sentence through a tokenized *MSA* and a tokenized *EDA* 5-grams LMs to get sentence perplexity from each LM (*msaPPL* and *edaPPL*). These two LMs are built using the same data and the same procedure for the LMs used in the "Token Level Identification" module;
- *Dia-Statistics-Features [DSF]*:
  - The percentage of words tagged as *EDA* in the input sentence by the "Token Level Identification" module (*diaPercent*);
  - The percentage of words tagged as *EDA* and *MSA* by *MADAMIRA* in the input sentence (*calimaWords* and *samaWords*, respectively). And the percentage of words found in a precompiled *EDA* lexicon *egyWords* used and provided by (Elfardy and Diab, 2013);
  - *hasUnk* is a binary feature set to "true" only if the language model of the "Token Level Identification" module yielded at least one *unk* tag in the input sentence;
  - Modality features: The percentage of words tagged as *EDA*, *MSA*, and both (*modEDA*, *modMSA*, and *modBoth*, respectively) using the *Modality List* component in the "Token Level Identification" module.
- *Sentence-Statistics-Features [SSF]*: The percentage of Latin words, numbers, and punctuation (*latinPercent*, *numPercent*, and *puncPercent*, respectively) in the input sentence. In addition to the average word length (*avgWordLen*) and the total number of words (*sentLength*) in the same sentence;
- *Sentence-decoration-features [SDF]*: Some binary features of whether the sentence has/doesn't have diacritics (*hasDiac*), speech effects (*hasSpeechEff*), presence of exclamation mark (*hasExMark*), presence of emoticons (*hasEmot*), presence of question mark (*hasQuesMark*), presence of decoration effects (*hasDecEff*) (ex: ****), or repeated punctuation (*hasRepPunc*).

3.2.2 Abstract Classifier

The second classifier, *Abs-Cl*, is intended to cover the implicit semantic and syntactic relations between words. It runs the input sentence in its surface form without tokenization through a surface form *MSA* and a surface form *EDA* 5-gram LMs to get sentence probability from each of the respective LM (*msaProb* and *edaProb*). These two LMs are built using the same data used in the "Token Level Identification" module LM, but without tokenization.

This classifier complements the information provided by *Comp-Cl*. While *Comp-Cl* yields detailed and specific information about the tokens as it uses tokenized-level LMs, *Abs-Cl* is able to capture better semantic and syntactic relations between words since it can see longer context in terms of the number of words compared to that seen by *Comp-Cl* (on average a span of two words in the surface-level LM corresponds to almost five words in the tokenized-level LM) (Rashwan et al., 2011).

3.2.3 DT Ensemble

In the final step, we use the classes and confidence scores of the preceding two classifiers on the training data to train a decision tree classifier. Accordingly, an input test sentence goes through *Comp-Cl* and *Abs-Cl*, where each classifier assigns the sentence a label and a confidence score for this label. It then uses the two labels and the two confidence scores to provide its final classification for the input sentence.

4 Experimental Setup

4.1 Data

To our knowledge, there is no publicly available standard dataset that is annotated for both token and sentence levels to be used for evaluating both levels of classifications. Accordingly we use two separate standard datasets for both tasks.

For the token level identification, we use the training and test data that is provided by the shared task ShTk. Additionally, we manually annotate more token-level data using the same guidelines used to annotate this dataset and use this additional data for training and tuning our system.

- *tokTrnDB*: is the ShTk training set. It consists of 119,326 words collected from Twitter;

- *tokTstDB*: is the ShTk test set. It consists of 87,373 words of tweets collected from some unseen users in the training set and 12,017 words of sentences collected from Arabic commentaries;
- *tokDevDB*: 42,245 words collected from weblogs and manually annotated in house using the same guidelines of the shared task.[7] We only use this set for system tuning to decide upon the best configuration;
- *tokTrnDB2*: 171,419 words collected from weblogs and manually annotated in house using the same guidelines of the shared task. We use it as an extra training set in addition to *tokTrnDB* to study the effect of increasing training data size on the system performance.[8]

Table 1 shows the distribution of each of these subsets of the token-level dataset.

	lang1	*lang2*	*ambig*	*ne*	*other*	*mixed*
tokTrnDB	79,059	16,291	1,066	14,110	8,688	15
tokTstDB	57,740	21,871	240	11,412	8,121	6
tokTrnDB2	77,856	69,407	46	14,902	9,190	18
tokDevDB	23,733	11,542	34	4,017	2,916	3

Table 1: Tag distribution in the datasets used in our token level identification component.

For sentence level dialect identification, we use the code-switched *EDA-MSA* portion of the crowd source annotated dataset (Zaidan and Callison-Burch, 2011). The dataset consists of user commentaries on Egyptian news articles. The data is split into training (*sentTrnDB*) and test (*sentTstDB*) using the same split reported by Elfardy and Diab (2013). Table 2 shows the statistics for that data.

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

Table 2: Number of *EDA* and *MSA* sentences and tokens in the training and test sets.

4.2 Baselines

4.2.1 Token Level Baselines

For the token level task, we evaluate our approach against the results reported by all systems participating in ShTk evaluation test bed. These baselines include:

- *IUCL*: The best results obtained by King et al. (2014);
- *IIIT*: The best results obtained by Jain and Bhat (2014);
- *CMU*: The best results obtained by Lin et al. (2014);
- *MSR-India*: The best results obtained by Chittaranjan et al. (2014);
- *AIDA*: The best results obtained by us using the older version *AIDA* (Elfardy et al., 2014).

4.2.2 Sentence Level Baselines

For the sentence level component, we evaluate our approach against all published results on the Arabic "Online Commentaries (AOC)" publicly available dataset (Zaidan and Callison-Burch, 2011). The sentence level baselines include:

- *Zidan et al*: The best results obtained by Zaidan and Callison-Burch (2011);
- *Elfardy et al*: The best results obtained by Elfardy and Diab (2013);
- *Cotterell et al*: The best result obtained by Cotterell and Callison-Burch (2014a);
- *All Features*: This baseline combines all features from *Comp-Cl* and *Abs-Cl* to train a single decision tree classifier.

5 Evaluation

5.1 Token Level Evaluation

Table 3 compares our token level identification approach to all baselines. It shows, our proposed approach significantly outperforms all baselines using the same training and test sets. *AIDA*2 achieves 90.6% weighted average F-score while the nearest baseline gets 86.8% (this is 28.8% error reduction from the best published approach). By using both *tokTrnDB* and *tokTrnDB2* for training, the weighted average F-score is further improved by 2.3% as shown in the last row of the table.

5.2 Sentence Level Evaluation

For all experiments, we use a decision-tree classifier as implemented in *WEKA* (Hall et al., 2009) toolkit. Table 4 shows the 10-folds cross-validation results on the *sentTrnDB*.

- "*Comp-Cl*" shows the results of the best selected set of features from each group. (The

[7] The task organizers kindly provided the guidelines for the task.

[8] We are expecting to release both *tokDevDB* and *tokTrnDB2* in addition to some other data are still under development to the community by 2016

Baseline	*lang1*	*lang2*	*ambig*	*ne*	*other*	*mixed*	Avg-F
AIDA	89.4	76.0	0.0	87.9	99.0	0.0	86.8
CMU	89.9	81.1	0.0	72.5	98.1	0.0	86.4
IIIT	86.2	52.9	0.0	70.1	84.2	0.0	76.6
IUCL	81.1	59.5	0.0	5.8	1.2	0.0	61.0
MSR-India	86.0	56.4	0.7	49.6	74.8	0.0	74.2
*AIDA***2**	**92.9**	**82.9**	**0.0**	**89.5**	**99.3**	**0.0**	**90.6**
*AIDA***2+**	**94.6**	**88.3**	**0.0**	**90.2**	**99.4**	**0.0**	**92.9**

Table 3: F-score on held-out test-set *tokTstDB* using our best setup against the baselines. *AIDA*2+ shows the the results of training our system using *tokTrnDB* and *tokTrnDB2*

	Group	Accuracy
Comp-Cl	**Perplexity-Features**	80.0%
	Dia-Statistics-Features	85.1%
	Sentence-Statistics-Features	61.6%
	Sentence-decoration-features	53.1%
	Best of all groups	87.3%
Abs-Cl		78.4%
DT Ensemble		89.9%

Table 4: Cross-validation accuracy on the *sentTrnDB* using the best selected features in each group

ones that yield best cross-validation results of *sentTrnDB*. "Best-of-all-groups" shows the result of the best selected features from the retained feature groups which in turn is the final set of features for the comprehensive classifier. In our case the best selected features are *msaPPL*, *edaPPL*, *diaPercent*, *hasUnk*, *calimaWords*, *modEDA*, *egyWords*, *latinPercent*, *puncPercent*, *avgWordLen*, and *hasDiac*.

- "*Abs-Cl*" shows the results and best set of features (*msaProb* and *edaProb*) for the abstract classifier.
- "DT Ensemble" reflect the results of combining the labels and confidence scores from *Comp-Cl* and *Abs-Cl* using a decision tree classifier.

Among the different configurations, the ensemble system yields the best 10-fold cross-validation accuracy of 89.9%. We compare the performance of this best setup to our baselines on both the cross-validation and held-out test sets. As Table 5 shows, the proposed approach significantly outperforms all baselines on all sets.

6 Results Discussion

6.1 Token Level Results Discussion

Last row in table 3 shows that the system results in 24.5% error reduction by adding 171K words

Baseline	*sentTrnDB*	*sentTstDB*	*sentTrnDB + sentTstDB*
Zidan et al	N/A	N/A	80.9
Elfardy et al	85.3	83.3	85.5
Cotterell et al	N/A	N/A	86.6
All Features	85.8	85.3	85.5
DT Ensemble	**89.9**	**87.3**	**90.8**

Table 5: Results of using our best setup (DT Ensemble) against baselines

of gold data to the training set. This shows that the system did not reach the saturation state yet, which means that adding more gold data can increase performance.

Table 6 shows the confusion matrix of our best setup for all six labels over the *tokTstDB*. The table shows that the highest confusability is between *lang1* and *lang2* classes; 2.9% are classified as *lang1* instead of *lang2* and 1.6% are classified as *lang2* instead of *lang1*. This accounts for 63.8% of the total errors. The Table also shows that our system does not produce the *mixed* class at all probably because of the tiny number of *mixed* cases in the training data (only 33 words out of 270.7K words). The same case applies to the *ambig* class as it represents only 0.4% of the whole training data. *lang1* and *ne* are also quite highly confusable. Most of *ne* words have another non-named entity meaning and in most cases these other meanings tend to be *MSA*. Therefore, we expect that a more sophisticated NER system will help in identifying these cases.

		Predicted					
		lang1	*lang2*	*ambig*	*ne*	*other*	*mixed*
Gold	*lang1*	55.7%	1.6%	0.0%	0.9%	0.0%	0.0%
	lang2	2.9%	18.9%	0.0%	0.2%	0.0%	0.0%
	ambig	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%
	ne	0.8%	0.2%	0.0%	10.3%	0.1%	0.0%
	other	0.0%	0.0%	0.0%	0.0%	8.2%	0.0%
	mixed	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 6: Token-level confusion matrix for the best performing setup on *tokTstDB*

Table 7 shows examples of the words that are misclassified by our system. The misclassified word in the first examples (bED meaning "each other") has a gold class *other*. However, the gold label is incorrect and our system predicted it correctly as *lang2* given the context. In the second example, the misclassified named entity refers to the name of a charitable organization but the word also means "message" which is a *lang1* word. The third example shows a *lang1* word that is incorrectly classified by our system as *lang2*. Similarly,

in the last example our system incorrectly classified a *lang2* word as a *lang1*.

Sentence	Word	Gold	Pred
tlt twytr mzwr . nSh AkwntAt m$ $gAlh w AlbAqy mblkyn bED تلت تويتر مزور . نصه اكونتات مش شغالة و الباقي مبلكين بعض *One third of twitter is forged. Half of the accounts are not working while the rest block each other..*	bED بعض each other	*other*	*lang2*
kmA Anny mTlE Ely mA tqwmwn bh fy mxtlf AljmEyAt wAlAn$Tp AlAhlyp . mvl rsAlp . كما اني مطلع علي ما تقومون به في مختلف الجمعيات والانشطة الاهلية . مثل رسالة. *Also I know what you are doing in different domains and civil activities like Resala.*	rsAlp رسالة Resala	*ne*	*lang1*
>nA bxyr . SHty wAlHmd llh fy >fDl HAl . انا بخير صحتي والحمد لله في افضل حال. *I am fine. Thank God, my health is in best condition.*	SHty صحتي my health	*lang1*	*lang2*
lm Aqr> AlbyAn w qrrt AEtrD Elyh glAsp لم اقرأ البيان و قرّرت اعترض عليه غلاسة *I did not read the statement and decided to object to it just to be annoying*	AEtrD اعترض object	*lang2*	*lang1*

Table 7: Examples of the words that were misclassified by our system

6.2 Sentence Level Results Discussion

The best selected features shows that *Comp-Cl* benefits most from using only 11 features. By studying the excluded features we found that:

- Five features (*hasSpeechEff*, *hasEmot*, *hasDecEff*, *hasExMark*, and *hasQuesMark*) are zeros for most records, hence extremely sparse, which explains why they are not selected as relevant distinctive features. However, it should be noted that the *hasSpeechEff* and *hasEmot* features are markers of informal language especially in the social media (not to ignore the fact that users write in *MSA* using these features as well but much less frequently). Accordingly we anticipate that if the data has more of these features, they would have significant impact on modeling the phenomena;
- Five features are not strong indicators of dialectalness. For the *sentLength* feature, the average length of the *MSA*, and *EDA* sentences in the training data is almost the same. While, the *numPercent*, *modMSA*, *modBoth*, and *hasRepPunc* features are almost uniformly distributed across the two classes;
- The initial assumption was that *SAMA* is exclusively *MSA* while *CALIMA* is exclusively *EDA*, thereby the *samaWords* feature will be a strong indicator for *MSA* sentences and the *calimaWords* feature will be a strong indicator for *EDA* sentences. Yet by closer inspection, we found that in 96.5% of the *EDA* sentences, *calimaWords* is higher than *samaWords*. But, in only 23.6% of the *MSA* sentences, *samaWords* is higher than *calimaWords*. This means that *samaWords* feature is not able to distinguish the *MSA* sentences efficiently. Accordingly samaWords feature was not selected as a distinctive feature in the final feature selection process.

Although *modEDA* is selected as one of the representative features, it only occurs in a small percentage of the training data (10% of the *EDA* sentences and 1% of the *MSA* sentences). Accordingly, we repeated the best setup (DT Ensemble) without the modality features, as an ablation study, to measure the impact of modality features on the performance. In the 10-fold-cross-validation on the *sentTrnDB* using *Comp-Cl* alone, we note that performance results slightly decreased (from 87.3% to 87.0%). However given the sparsity of the feature (it occurs in less than 1% of the tokens in the *EDA* sentences), 0.3% drop in performance is significant. This shows that if the modality lexicon has more coverage, we will observe a more significant impact.

Table 8 shows some examples for our system predictions. The first example is correctly classified with a high confidence (92%). Example 2 is quite challenging. The second word is a typo where two words are concatenated due to a missing white space, while the first and third words can be used in both *MSA* and *EDA* contexts. Therefore, the system gives a wrong prediction with a low confidence score (59%). In principle this sentence could be either *EDA* or *MSA*. The last example should be tagged as *EDA*. However, our system tagged it as *MSA* with a very high confidence score of (94%).

Input sentence	Gold	Pred	Conf
wlA AElAnAt fY Altlyfzywn nAfEp w lA jrArAt jdydp nAfEp.. w bEdyn. ولا اعلانات في التلفزيون نافعة ولا جرارات جديدة نافعة.. وبعدين. *Neither TV commercials nor new tractors work. So now what.*	*EDA*	*EDA*	92%
Allhm AgfrlhA wArHmhA اللّهم اغفرلها وارحمها *May God forgive her and have mercy on her.*	*MSA*	*EDA*	59%
tsmHly >qwlk yAbAbA? تسمحلي اقولك يابابا؟ *Do you allow me to call you father?*	*EDA*	*MSA*	94%

Table 8: Examples of the sentences that were misclassified by our system

7 Conclusion

We presented *AIDA*2, a hybrid system for token and sentence levels dialectal identification in code switched Modern Standard and Egyptian Dialectal Arabic text. The proposed system uses a classifier ensemble approach to perform dialect identification on both levels. In the token level module, we run the input sentence through four different classifiers. Each of which classify each word in the sentence. A CRF model is then used to predict the final class of each word using the provided information from the underlying four classifiers. The output from the token level module is then used to train one of the two underlying classifiers of the sentence level module. A decision tree classifier is then used to to predict the final label of any new input sentence using the predictions and confidence scores of two underlying classifiers. The sentence level module also uses a heuristic features selection approach to select the best features for each of its two underlying classifiers by maximizing the accuracy on a cross-validation set. Our approach significantly outperforms all published systems on the same training and test sets. We achieve 90.6% weighted average F-score on the token level identification compared to 86.8% for state of the art using the same data sets. Adding more training data results in even better performance to 92.9%. On the sentence level, *AIDA*2 yields an accuracy of 90.8% using cross-validation compared to the latest state of the art performance of 86.6% on the same data.

References

Rania Al-Sabbagh, Jana Diesner, and Roxana Girju. 2013. Using the semantic-syntactic interface for reliable arabic modality annotation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 410–418, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

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.

Yassine Benajiba, Paolo Rosso, and Jos Miguel Benedruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *In Proceedings of CICLing-2007.*

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.*

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. 2014a. A multi-dialect, multi-genre corpus of informal written arabic. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Ryan Cotterell and Chris Callison-Burch. 2014b. A multi-dialect, multi-genre corpus of informal written arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Heba Elfardy and Mona Diab. 2013. Sentence-Level Dialect Identification in Arabic. In *Proceedings of ACL2013*, Sofia, Bulgaria, August.

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

Jr. Forney, G.D. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. *Proceedings of the 7th Meeting of the North AmericanChapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06)*.

Nizar Habash, Ramy Eskander, and AbdelAti Hawwari. 2012. A Morphological Analyzer for Egyptian Arabic. In *NAACL-HLT 2012 Workshop on Computational Morphology and Phonology (SIGMORPHON2012)*, pages 1–9.

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).

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.

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.

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.

Mohamed Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. Ldc standard arabic morphological analyzer (sama) version 3.1.

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.

M.A.A. Rashwan, M.A.S.A.A. Al-Badrashiny, M. Attia, S.M. Abdou, and A. Rafea. 2011. A stochastic arabic diacritizer based on a hybrid of factorized and unfactorized textual features. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):166–175, Jan.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence level dialect identification for machine translation system selection. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. pages 213–220.

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 *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*.

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