Improved Sentence-Level Arabic Dialect Classification

Christoph Tillmann and **Yaser Al-Onaizan**
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{ctill,onaizan}@us.ibm.com

Saab Mansour*
Aachen University
Aachen, Germany
mansour@cs.rwth-aachen.de

Abstract

The paper presents work on improved sentence-level dialect classification of Egyptian Arabic (ARZ) vs. Modern Standard Arabic (MSA). Our approach is based on binary feature functions that can be implemented with a minimal amount of task-specific knowledge. We train a feature-rich linear classifier based on a linear support-vector machine (linear SVM) approach. Our best system achieves an accuracy of 89.1 % on the Arabic Online Commentary (AOC) dataset (Zaidan and Callison-Burch, 2011) using 10-fold stratified cross validation: a 1.3 % absolute accuracy improvement over the results published by (Zaidan and Callison-Burch, 2014). We also evaluate the classifier on dialect data from an additional data source. Here, we find that features which measure the informalness of a sentence actually decrease classification accuracy significantly.

1 Introduction

The standard form of written Arabic is Modern Standard Arabic (MSA) . It differs significantly from various spoken varieties of Arabic (Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014; Elfardy and Diab, 2013). Even though these dialects do not originally exist in written form, they are present in social media texts. Recently a dataset of dialectal Arabic has been made available in the form of the **Arabic Online Commentary** (AOC) set (Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014). The data consists of reader commentary from the online versions of Arabic newspapers, which have a high degree of dialect content. Data for the following dialects has been collected: Levantine, Gulf, and Egyptian. The data had been obtained by a crowd-sourcing effort. In the current paper, we present results for a binary classification task only, where we predict the dialect of Egyptian Arabic ARZ vs. MSA sentences from the *Al-Youm Al-Sabe'* newspaper online commentaries [1]. Our ultimate goal is to use the dialect classifier for building a dialect-aware Arabic-English statistical machine translation (SMT) system. Our Arabic-English training data contains a significant amount of Egyptian dialect data only, and we would like to adapt the components of our hierarchical phrase-based SMT system (Zhao and Al-Onaizan, 2008) to that data.

Similar to (Elfardy and Diab, 2013), we present a sentence-level classifier that is trained in a supervised manner. Our approach is based on an Arabic tokenizer, but we do not use a range of specialized tokenizers or orthography normalizers. In contrast to the language-model (LM) based classifier used by (Zaidan and Callison-Burch, 2014), we present a linear classifier approach that works best without the use of LM-based features. Some improvements in terms of classification accuracy and 10-fold cross validation under the same data conditions as (Zaidan and Callison-Burch, 2011; Elfardy and Diab, 2013) are presented. In general, we aim at a smaller amount of domain specific feature engineering than previous related approaches.

The paper is structured as follows. In Section 2, we present related work on language and dialect identification. In Section 3, we discuss the linear classification model used in this paper. In Section 4, we evaluate the classifier performance in terms of classification accuracy on two data sets and present some

*Part of the work was done while the author was a student intern at the IBM T.J. Watson Research Center.

[1] We use the ISO 639-3 code ARZ for denoting Egyptian Arabic.

Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects, pages 110–119,
Dublin, Ireland, August 23 2014.

error analysis. Finally, in Section 5, we discuss future work on improved dialect-level classification and its application to system adaptation for machine translation.

2 Related Work

From a computational perpective, we can view dialect identification as a more fine-grained form of language identification (ID). Previous work on language ID examined the use of character histograms (Cavnar and Trenkle, 1994; Dunning, 1994), and high accuracy prediction results have been reported even for languages with a common character set. (Baldwin and Lui, 2010) present a range of document-level language identification techniques on three different data sets. They use n-gram counting techniques and different tokenization schemes that are adopted to those data sets. Their classification task deals with several languages, and it becomes more difficult as the number of languages increases. They present an SVM-based multiclass classification approach similar to the one presented in this paper which performs well on one of their data sets. (Trieschnigg et al., 2012) generates n-gram features based on character or word sequences to classify dialectal documents in a dutch-language fairy-tale collection. Their baseline model uses N-gram based text classification techniques as popularised in the *TextCat* tool (Cavnar and Trenkle, 1994). Following (Baldwin and Lui, 2010), the authors extend the usage of n-gram features with nearest neighbour and nearest-prototype models together with appropriately chosen similarity metrics. (Zampieri and Gebre, 2012) classify two varieties of the same language: European and Brazilian Portuguese. They use word and character-based language model classification techniques similar to (Zaidan and Callison-Burch, 2014). (Huang and Lee, 2008) present simple bag-of-word techniques to classify varieties of Chinese from the Chinese Gigaword corpus. (Kruengkrai et al., 2005) extend the use of n-gram features to using string kernels: they may take into account all possible sub-strings for comparison purposes. The resulting kernel-based classifier is compared against the method in (Cavnar and Trenkle, 1994). (Lui and Cook, 2013) present a dialect classification approach to identify Australian, British, and Canadian English. They present results where they draw training and test data from different sources. The successful transfer of models from one text source to another is evidence that their classifier indeed captures dialectal rather than stylistic or formal differences. Language identification of related languages is also addressed in the DSL (Discriminating Similar Languages) task of the present Vardial workshop at COLING 14 (Tan et al., 2014).

While most of the above work focuses on document-level language classification, recent work on handling Arabic dialect data addresses the problem of sentence-level classification (Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014; Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2014). The work is based on the data collection effort by (Zaidan and Callison-Burch, 2014) which crowdsources the annotation task to workers on Amazons Mechanical Turk. The classification results by (Zaidan and Callison-Burch, 2014) are based on n-gram language-models, where the n-grams are defined both on words and characters. The authors find that unigram word-based models perform best. The word-based models are obtained after a minimal amount of preprocessing such as proper handling of HTML entities and Arabic numbers. Classification accuracy is significantly reduced for shorter sentences. (Elfardy and Diab, 2013) presents classifcation result based on various tokinization and orthographic normalization techniques as well as so-called *meta* features that estimate the informalness of the data. Like our work, the authors focus on a binary dialect classification based on the ARZ-MSA portion of the dataset in (Zaidan and Callison-Burch, 2011).

3 Classification Model

We use a linear model and compute a score $s(t_1^n)$ for a tokenized input sentence consisting of n tokens t_i:

$$s(t_1^n) \;=\; \sum_{s=1}^{d} w_s \cdot \sum_{i=1}^{n} \phi_s(c_i, t_i) \tag{1}$$

where $\phi_s(c_i, t_i)$ is a binary feature function which takes into account the context c_i of token t_i. $\mathbf{w} \in \mathbb{R}^d$ is a high-dimensional weight vector obtained during training. In our experiments, we classify a tokenized

Description	MSA		ARZ	
	# sentences	# words	# sentences	# words
ARZ-MSA portion of AOC	$13,512$	$334K$	$12,527$	$327K$
DEV12 tune set	585	$8.4K$	634	$9.3K$

Table 1: We used the following dialect data: 1) the ARZ-MSA portion of the AOC data from commentaries of the Egyptian newspaper Al-Youm Al-Sabe', and 2) the DEV12 tune set (1219 sentences) which is the LDC2012E30 corpus BOLT Phase 1 dev-tune set. The DEV12 tune set was annotated by a native speaker of Arabic.

sentence as being Egyptian dialect (ARZ) if $s(t_1^n) > 0$. To train the weights $\mathbf{w}$ in Eq. 1, we use a linear SVM approach (Hsieh et al., 2008; Fan et al., 2008). The trainer can easily handle a huge number of instances and features. The training data is given as instance-label pairs (x_i, y_i) where $i \in \{1, \cdots, l\}$ and l is the number of training sentences. The x_i are d-dimensional vectors of integer-valued features that count how often a binary feature fired for a tokenized sentence t_1^n. $y_i \in \{+1, -1\}$ are the class labels where a label of '+1' represents Egyptian dialect. During training, we solve the following optimization problem:

$$\min_w ||\mathbf{w}||_1 \; + \; C \sum_{i=1}^{l} \max(0, 1 - y_i \, \mathbf{w}^{\mathbf{T}} \, x_i \,) \; , \tag{2}$$

i.e. we use $L1$ regularized $L2$-loss support vector classification. We set the penalty term $C = 0.5$. For our experiments, we use the data set provided in (Zaidan and Callison-Burch, 2011) which also has been used in the experiments in (Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2014). We focus on the binary classification between MSA and ARZ. Details on the data sources can be found in Table 1. We present accuracy results in terms of 10-fold stratified cross-validation which are comparable to previously published work.

3.1 Tokenization and Dictionaries

The Arabic tokenizer used in the current paper is based on (Lee et al., 2003). It is a general purpose tokenizer which has been optimized towards improving machine translation quality of SMT systems rather than dialect classification. Together with the tokenized text, a maximum-entropy based tagger provides the part-of-speech (PoS) tags for each token. In addition, we have explored a range of features that are based on the output of the AIDA software package (Elfardy and Diab, 2012; Mona Diab et al., 2009 2011). The AIDA software has been made available to the participants of the DARPA-funded Broad Operational Language Translation (BOLT) project. AIDA is a system for dialect identification, classification and glossing on the token and sentence level for written Arabic. AIDA aggregates several components including dictionaries and language models in order to perform named entity recognition, dialect identification classification, and MSA English linearized glossing of the input text. We created a dictionary from AIDA resources that includes about $41\,000$ ARZ tokens. In addition, we obtained a second small dictionary of about 70 ARZ dialect tokens with the help of a native speaker of Arabic. The list was created by training two IBM Model 1 lexicons, one on Egyptian Arabic data and another on MSA data. We then inspected the ARZ lexicon entries with the highest cosine distance to their MSA counterparts and kept the ones that are strong ARZ words. The tokens in both dictionaries are not ARZ exclusive, but could occur in MSA as well.

3.2 Feature Set

In our work, we employ a simple set of binary feature functions based on the tokenized Arabic sentence. For example, we define a token bigram feature as follows:

$$\phi_{Bi}(t_k, t_{k-1}) \quad = \quad \begin{cases} 1 & t_k = \text{'قوي'} \text{ and } t_{k-1} = \text{'حلو'} \\ 0 & \text{otherwise} \end{cases} \; . \tag{3}$$

Token unigram and trigram features are defined accordingly. We also define unigram, bigram, and trigram features based on PoS tags. Currently, just PoS unigrams are used in the experiments. We define dictionary-based features as follows:

$$\phi_{Dict_l}(t_k) \quad = \quad \begin{cases} 1 & t_k = \text{`دلوقت'} \text{ and } t_k \in Dict_l \\ 0 & \text{otherwise} \end{cases} , \tag{4}$$

where we use the two dictionaries $Dict_1$ and $Dict_2$ as described in Section 3.1. The dictionaries are handled as token sets and we generate separate features for each of them. We generate some features based on the AIDA tool output. AIDA provides a dialect label for each input token t_k as well as a single dialect label at the sentence level. A sentence-level binary feature based on the AIDA sentence level classification is defined as follows:

$$\phi_{\text{AIDA}}(t_1^n) \quad = \quad \begin{cases} 1 & AIDA(t_1^n) \text{ is ARZ} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $AIDA(t_1^n)$ is the sentence-level classification of the AIDA tool. A word-level feature $\phi_{\text{AIDA}}(t_k)$ is defined accordingly. These features improve the classification accuracy of our best system significantly.

We have also experimented with some real-valued feature. For example, we derived a feature from dialect-specific language model probabilities:

$$\phi_{LM}(t_1^n) \quad = \quad 1/n \cdot [\, log(p_{\text{MSA}}(t_1^n)) \; - \; log(p_{\text{ARZ}}(t_1^n))] \,,$$

where $log(p_{\text{ARZ}}(t_1^n))$ is the language-model log probability for the dialect class ARZ . We used a trigram language model. $p_{\text{MSA}}(\cdot)$ is defined accordingly. In addition, we have implemented a range of so-called 'meta' features similar to the ones defined in (Elfardy and Diab, 2013). For example, we define a feature $\phi_{Excl}(t_1^n)$ which is equal to the length of the longest consecutive sequence of exclamation marks in the tokenized sentence t_1^n . Similarly, we define features that count the longest sequence of punctuation marks, the number of tokens, the averaged character-length of a token in the sentence, and the percentage of words with word-lengthening effects. These features do not directly model dialectalness of the data but rather try to capture the degree of in-formalness. Contrary to (Elfardy and Diab, 2013) we find that those features do not improve accuracy of our best model in the cross-validation experiments. On the DEV12 set, the use of the meta features results in a significant drop in accuracy.

4 Experiments

In this section, we present experimental results. Firstly, Section 4.1 demonstrates that our data is annotated consistently. In Section 4.2, we present dialect prediction results in terms of accuracy and F-score on our two data sets. In Section 4.3, we perform some qualitative error analysis for our classifier. In Section 4.4, we present some preliminary effects on training a SMT system.

4.1 Annotator Agreement

To confirm the consistent annotation of our data, we have measured some inter-annotator and intra-annotator agreement on it. A native speaker of Arabic was asked to classify the ARZ-MSA portion of the dialect data using the following three labels: ARZ, MSA, Other. We randomly sampled 250 sentences from the ARZ-MSA portion of the Zaidan data maintaining the original dialect distribution. The confusion matrix is shown in Table 2. It corresponds to a kappa value of 0.84 (using the definition of (Fleiss, 1971)), which indicates a very high agreement. In addition, we did re-annotate a sub-set of 200 sentences from the DEV12 set over a time period of three months using our own annotator. The kappa value of the corresponding confusion matrix is 0.93, indicating very high agreement as well.

4.2 Classification Experiments

Following previous work, we present dialect prediction results in terms of accuracy:

$$\textbf{ACC} \quad = \quad \frac{\text{\# sent correctly tagged}}{\text{\# sent}} \,, \tag{6}$$

		Predicted Class (IBM)		
		ARZ	MSA	Other
Actual	ARZ	125	4	1
Class	MSA	14	105	1
(AOC)	Other	0	0	0

Table 2: Inter annotator agreement on 250 randomly selected AOC sentences from the data in Table 1. An in-lab annotator's dialect prediction is compared against the AOC data gold-standard dialect labels.

where '# sent' is the number of sentences. In addition, we present dialect prediction results in terms of precision, recall, and F-score. They are defined as follows:

$$\begin{aligned} \mathbf{Prec} &= \frac{\text{\# sent correctly tagged as ARZ}}{\text{\# sent tagged as ARZ}} \\ \mathbf{Recall} &= \frac{\text{\# sent correctly tagged as ARZ}}{\text{\# ref sent tagged as ARZ}} \\ \mathbf{F} &= \frac{2 \cdot Prec \cdot Recall}{(Prec + Recall)}. \end{aligned} \tag{7}$$

MSA prediction F-score is defined analogously. Experimental results are presented in Table 3, where we present results for different sets of feature types and the two test sets in Table 1. In the top half of the table, results are presented in terms of 10-fold cross validation on the ARZ-MSA portion of the AOC data. In the bottom half, we present results on DEV12 tune set, where we use the entire dialect data in Table 1 for training (about $26K$ sentences).

As our baseline we have re-implemented the language-model-perplexity based approach reported in (Zaidan and Callison-Burch, 2011). We train language models on the dialect-labeled commentary training data for each of the dialect classes $c \in \{\text{MSA}, \text{ARZ}\}$. During testing, we compute the language model probability of a sentence s for each of the classes c. We assign a sentence to the class c with the highest probability (or the lowest perplexity) . For the 10-fold cross validation experiments, 10 language models are built and perplexities are computed on 10 different test sets. The resulting (averaged) accuracy is 83.3 % for cross-validation and 82.2 % on the DEV12 tune set. In comparison, (Elfardy and Diab, 2013) reports an accuracy of 80.4 % as perplexity-based baseline. We have carried out additional experiments with a simple feature set that consists of only unigram token and bigram token features as defined in Eq. 3. Such a system performs surprisingly well under both testing conditions: we achieved an accuracy of 87.7 % on the AOC data and an accuracy of 83.4 % on the DEV12 test set. On the AOC set using 10-fold cross validation, we achieve only a small improvement from using the dictionary features defined in Eq. 4. The accuracy is improved from 87.7 % to 88.0 %. On the DEV12 set, we obtain a much larger improvement from using these features. Furthermore, we have investigated the usefulness of the AIDA-based features. The stand-alone sentence-level classification of the AIDA tool performs quite poorly. On the DEV12 set, it achieves an accuracy of just 77.9 %. But using the AIDA assigned sentence-level and token-level dialect labels based on the binary features defined in Eq. 5 improves accuracy significantly, e.g. from 85.3 % to 87.8 % on the DEV12 set. In the current experiments, the so-called meta features which are computed at the sentence level do not improve classification accuracy. The meta features are only useful in classifying dialect data based on the in-formalness of the data, i.e. the ARZ news commentaries tend to exhibit more in-formalness than the MSA commentaries. Finally, the sentence-level perplexity feature defined in Eq. 6 did not improve accuracy as well (no results for this feature are presented in Table 3).

4.3 Classifier Analysis

In this section, we perform a simple error analysis of the classifier performance on some dialect data for which the degree of dialectalness is known. The data comes from news sources that differ from the data used to train the classifier. The classifier is evaluated on data from the DARPA-funded BOLT project.

	Feature Types		MSA			ARZ		
		ACC [%]	PREC	REC	F	PREC	REC	F
10-fold AOC	language-model	83.3	86.7	90.2	88.4	89.0	85.0	86.9
	aida-sentence label	81.0	84.2	78.0	81.0	78.0	84.3	81.0
	uni,bi	87.7	86.6	90.2	88.4	89.0	85.0	86.9
	uni,bi,dict,pos	88.0	86.9	90.4	88.6	89.2	85.3	87.2
	uni,bi,dict,pos,aida	89.1	87.5	92.2	89.8	91.1	85.7	88.3
	uni,bi,dict,pos,aida,meta	88.8	87.4	91.7	89.5	90.6	85.7	88.1
DEV12	language-model	82.2	85.1	76.2	80.4	80.0	87.7	83.7
	aida-sentence label	77.9	80.9	70.8	75.5	75.8	84.5	79.9
	uni,bi	83.4	81.1	85.1	83.1	85.6	81.7	83.6
	uni,bi,dict,pos	85.3	83.5	87.5	85.5	88.0	84.1	86.0
	uni,bi,dict,pos,aida	87.8	83.4	93.0	88.0	92.8	83.0	87.6
	uni,bi,dict,pos,aida,meta	68.3	61.8	90.8	73.5	85.0	48.3	61.6

Table 3: Arabic Dialect Classification Results: predicting MSA vs. (ARZ) dialect in terms of 10-fold cross-validation on the AOC data and on the DEV12 set using all the AOC data for training.

Corpus	#Sent	#Sent [ARZ]	%[ARZ]
ARZ web forum	299K	183K	61%
Broadcast	169K	18K	11%
Newswire	885K	29K	3%

Table 4: Sub-corpora together with total number as well as percentage of sentences that are classified as ARZ.

The BOLT data consists of several corpora collected from various resources. These resources include newswire, web-logs, ARZ web forum data and others. Classification statistics are presented in Table 4, where we report the number of sentences along with the percentage of those sentences classified as ARZ. The distribution of the dialect labels in the classifier output appears to correspond to the expected origin of the data. For example, the ARZ web forum data contains a majority of ARZ sentences, but quite a few sentences are MSA such as greetings and quotations from Islamic resources (Quran, Hadith ...). The broadcast conversation data is mainly MSA, but sometimes the speaker switches to dialectal usage for a short phrase and then switches back to MSA. Lastly, the newswire data has a vast majority of MSA sentences. Examining a small portion of newswire sentences classified as ARZ, the sentences labeled as ARZ are mostly classification errors.

Example sentence classifications from the BOLT data are shown in Table 5. The first two text fragments are taken from the Egyptian Arabic (ARZ) web forum data. In the first document fragment, the user starts with MSA sentences, then switches to Egyptian (ARZ) dialect marked by the ARZ indicator اللي and using the prefix ب# before a verb which is not allowed in MSA. The user then switches back to MSA. The classifier is able to classify the Egyptian Arabic (ARZ) sentence correctly. In the second document fragment, the user uses several Egyptian Arabic (ARZ) words. In the forth sentence no ARZ words exist, and the classifier correctly classifies the sentence as MSA. The third text fragment shows

Predicted Dialect	Arabic	English
MSA	انا قرات الموضوع و الردود .	i read the topic and the replies .
MSA	الموضوع فكرة حلوة .	the topic is great !
ARZ	و # انا مع الاخ **اللي ب # يقول**	i agree with the brother **who said**
MSA	الدين مهم في كل حاجة	Islam is significant in all
ARZ	**علشان** الناس **دي** صبرت علي البلاء	**because they** accept affliction with patience
ARZ	و **اللي** عملت +ه حماس **دة** اكبر انتصار	**what** Hamas did was a victory
ARZ	**زي** حماس وقفوا في **وش** احتلال	who encountered the occupation
MSA	و صبروا علي حصار	and they were patient despite the siege
ARZ	**علشان كده** رب +نا كافئ +هم	**that 's why** Allah rewarded them
ARZ*	و # قد قادت تي **دي** كه	t**d**k ... led
ARZ*	و # ينحو خبراء النقل **ب #**اللائمة	transport experts blame
ARZ*	لا استطيع تذكر ما قال +ه ل # +ي .	i cannot remember what he told me

Table 5: Automatic classification examples for the dialect classes ARZ and MSA. Arabic source and English target sentences are given. Dialectal words are in **bold**. Incorrect predictions are marked by an asterisk (*).

some sentences from the newswire corpus that are mis-classified. The first sentence contains the word دي which corresponds to the letter 'd' in the abbreviation 'tdk'. The word is contained in one of our ARZ dictionaries such that the binary AIDA-based feature in Eq. 5 fires and triggers a mis-classification. In this context, the word is part of an abbreviation which is split in the Arabic text. In the other examples, only a few of the binary features defined in Section 3.2 apply and features that correspond to Arabic prefixes tend to support a classification as ARZ dialect.

4.4 Preliminary Application for SMT

The dialect classification of Arabic data for SMT can be used in various ways. Examples include domain-specific tuning, mixture modeling, and the use of so-called provenance features (Chiang et al., 2011) among others . As a motivation for the future use of the dialect classifier in SMT, we classify the BOLT bilingual training data into ARZ and MSA parts and examine the effect on the phrase table scores. Phrase translation pairs demonstrating the use of the classified training data are shown in Table 6. The ARZ web forum data is split into an ARZ part and an MSA part and two separate phrase probability tables are trained on these two splits. The ARZ web forum data is highly ambiguous with respect to dialect and it is difficult to obtain good dialect-dependent splits of the data. In the first example in the table, the word العربية could mean 'Arab' in MSA, but in ARZ it could also mean 'car'. The phrase table scores obtained from the classifier-split training data correctly reflect this ambiguity. The phrase pair with 'car' has the lowest translation score for the BOLT.ARZ phrase table, while it has a higher cost in the BOLT.MSA phrase table. In the full phrase table (BOLT), 'car' is the fifth translation candidate with a score of 2.09.

f	BOLT.ARZ e	cost	BOLT.MSA e	cost
العربية	the car	1.20	arab	0.80
	arab	1.25	the arab	1.32
	the arab	1.70	Arabic	1.52
مرسي	merci	1.53	marsa	1.99
	marsa	1.63	thanks	2.01
	mursi	1.91	morcy	2.13

Table 6: Phrase tables based on classified training data. BOLT.ARZ is trained on the ARZ portion of the ARZ web forums data, while BOLT.MSA is trained on the MSA part. The table includes Arabic words and the top three phrase translation candidates, sorted (first is best) by the phrase model cost (cost= $-log(p(f|e))$).

In the second example, the word مرسي could function as a proper noun with its English translation 'mursi' or 'marsa', but only in ARZ it could also be translated as 'thanks' ('merci'). In this case, the classifier is unable to distinguish between the ARZ dialect and the MSA usage. We found out that the word token 'merci' appears only 4 times in the training data, rendering its binary features unreliable reliable. In general we note that the phrase tables build on the classified data become more domain-specific, and it is left to future work to check whether improvements could carry over to the translation quality.

5 Discussion and Future Work

The ultimate goal is to use the ARZ vs. MSA dialect classifier for training an adapted SMT system. We split the training data at the sentence level using our classifier and train dialect-specific systems on each of these splits along with a general dialect-independent system. We will be using techniques similar to (Koehn and Schroeder, 2007; Chiang et al., 2011; Sennrich, 2012; Chen et al., 2013) to adapt the general SMT system to a target domain with a predominant dialect. Or, we will be adopting an SMT system to a development or test set where we use the classifier to predict the dialect for each sentence and use a dialect-specific SMT system on each of them individually. Our approach of using just binary feature functions in connection with a sentence-level global linear model can be related to work on PoS-tagging (Collins, 2002). (Collins, 2002) trains a linear model based on Viterbi decoding and the perceptron algorithm. The gold-standard PoS tags are given at the word-level, but the training uses a global representation at the sentence level. Similarly, we use linear SVMs (Hsieh et al., 2008) to train a classification model at the sentence level without access to sentence length statistics, i.e. our best performing classifier does not compute features like the percentage of punctuation, numbers, or averaged word length as has been proposed previously (Elfardy and Diab, 2013). All of our features are actually computed at the token level (with the exception of a single sentence-level AIDA-based feature). An interesting direction for future work could be to train the dialect classifier at the sentence level, but use it to compute token-level predictions for a more fine-grained analysis. Even though the token-level prediction task corresponds to a word-level tag set of just size 2, Viterbi decoding techniques could be used to introduce novel context-dependent features, e.g. dialect tag n-gram features. Such a token-level predictions might be used for weighting each phrase pair in an SMT system using methods like the instance-based adaptation approach in (Foster et al., 2010).

Acknowledgement

The current work has been funded through the Broad Operational Language Translation (BOLT) program under the project number DARPA HR0011-12-C-0015.

References

Timothy Baldwin and Marco Lui. 2010. Language Identification: The Long and the Short of the Matter. In *Proc. of HLT'10*, pages 229–237, Los Angeles, California, June.

William Cavnar and John M. Trenkle. 1994. N-gram-based Text Categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Boxing Chen, George Foster, and Roland Kuhn. 2013. Adaptation of reordering models for statistical machine translation. In *Proc. of HLT'13*, pages 938–946, Atlanta, Georgia, June.

David Chiang, Steve DeNeefe, and Michael Pust. 2011. Two Easy Improvements to Lexical Weighting. In *Proc. of HLT'11*, pages 455–460, Portland, Oregon, USA, June.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP'02*, pages 1–8, Philadelphia,PA, July.

Ted Dunning. 1994. Statistical Identification of Language. technical report mccs 94-273. Technical report, New Mexico State University.

Heba Elfardy and Mona Diab. 2012. Aida: Automatic Identification and Glossing of Dialectal Arabic. In *Proceedings of the 16th EAMT Conference (Project Papers)*, pages 83–83, Trento, Italy, May.

Heba Elfardy and Mona Diab. 2013. Sentence level dialect Identification in arabic. In *Proc. of the ACL 2013 (Volume 2: Short Papers)*, pages 456–461, Sofia, Bulgaria, August.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: a Library for Large Linear Classification. *Machine Learning Journal*, 9:1871–1874.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Proc. of EMNLP'10*, pages 451–459.

C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S.S. Keerthi, and S.Sundararajan. 2008. A Dual Coordinate Descent Method for Large-scale linear SVM. In *ICML*, pages 919–926, Helsinki,Finland.

Chu-Ren Huang and Lung-Hao Lee. 2008. Contrastive Approach towards Text Source Classication based on top-bag-of-word Similarity. In *PACLIC 2008*, pages 404–410, Cebu City, Philippines.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT07)*, pages 224–227.

Canasai Kruengkrai, Prapass Srichaivattana, Virach Sornlertlamvanich, and Hitoshi Isahara. 2005. Language Identification based on string kernels. In *In Proceedings of the 5th International Symposium on Communications and Information Technologies (ISCIT-2005*, pages 896–899.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. 2003. Language Model Based Arabic Word Segmentation. In *Proc. of the 41st Annual Conf. of the Association for Computational Linguistics (ACL 03)*, pages 399–406, Sapporo, Japan, July.

Marco Lui and Paul Cook. 2013. Classifying English Documents by National Dialect. In *Proc. Australasian Language Technology Workshop*, pages 5–15.

Mona Mona Diab, Heba Elfardy, and Yassine Benajiba. 2009–2011. AIDA Automatic Identification of Arabic Dialectal Text. a Tool for Dialect Identification & Classification, Named Entity Recognition, English and Modern Standard Arabic Glossing and Normalization.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proc. of EACL'12*, pages 539–549.

Liling Tan, Marcos Zampieri, Nicola Ljubešić, and Jörg Tiedemann. 2014. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *7th Workshop on Building and Using Comparable Corpora at LREC'14*, Reykjavik, Iceland, September.

D. Trieschnigg, D. Hiemstra, M. Theune F. Jong, and T. Meder. 2012. An Exploration of Language Identication Techniques for the Dutch Folktale Database. In *Adaptation of Language Resources and Tools for Processing Cultural Heritage Workshop (LREC 2012)*, Istanbul, Turkey, May.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proc. of ACL / HLT 11*, pages 1220–1229, Portland, Oregon, USA, June.

Omar F. Zaidan and Chris Callison-Burch. 2014. Arabic Dialect Classification. *CL*, 40(1):171–202.

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic Identication of Language Varieties: The case of Portuguese. In *Konvens 12*, pages 233–237, Vienna, Austria.

Bing Zhao and Yaser Al-Onaizan. 2008. Generalizing Local and Non-Local word-reordering patterns for syntax-based machine translation. In *Proc. of EMNLP'08*, pages 572–581, Honolulu, Hawaii, October.