Automatic Dialect Classification for Statistical Machine Translation

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

Yaser Al-Onaizan onaizan@us.ibm.com
Graeme Blackwood blackwood@us.ibm.com
Christoph Tillmann ctill@us.ibm.com
IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

Abstract

The training data for statistical machine translation are gathered from various sources representing a mixture of domains. In this work, we argue that when translating dialects representing varieties of the same language, a manually assigned data source is not a reliable indicator of the dialect. We resort to automatic dialect classification to refine the training corpora according to the different dialects and build improved dialect specific systems. A fairly standard classifier for Arabic developed within this work achieves state-of-the-art performance, with classification precision above 90%, making it usefully accurate for our application. The classification of the data is then used to distinguish between the different dialects, split the data accordingly, and utilize the new splits for several adaptation techniques. Performing translation experiments on a large scale dialectal Arabic to English translation task, our results show that the classifier generates better contrast between the dialects and achieves superior translation quality than using the original manual corpora splits.

1 Introduction

Training data for statistical machine translation (SMT) are extracted from various sources representing different domains (e.g., newswire, webforums, ...). The source of the data (encapsulated by meta-information) can be utilized to perform domain adaptation using different techniques. For example, mixture modeling of grammars trained on different sources of data (Foster and Kuhn, 2007), or provenance features using different sources of data (Chiang et al., 2011).

The meta-information based corpora split may contain further domain granularities. In this work, we tackle the case where the corpora contain a mixture of dialects. Dialects refer to varieties of a language, differing by vocabulary, morphology, grammar, etc. In this scenario, the meta-information split is rendered unreliable, and better splitting is required to achieve improvements using standard adaptation methods.

We start with developing an automatic dialect classifier for the purpose of refining the corpora splits. The classifier is applied on an Arabic dialect identification task, where we distinguish between the Egyptian Arabic (ARZ) and Modern Standard Arabic (MSA) dialects. MSA is the standard written form of Arabic while ARZ and other dialectal forms are mainly used for speech. Due to the prevalence of MSA in written form, most of the corpora collected for training SMT systems contain a majority of MSA data. Nevertheless, dialectal data has a strong presence in on-line content such as weblogs, forums and user commentary. Applying an automatic dialect classifier on corpora designated as dialectal shows that a large portion of the data is actually MSA, making dialectal identification essential for a successful utilization of the data.

Next, we extensively experiment with applying the classifier output for domain-adaptation, and compare using the classifier output to using meta-information based data splits. Various adaptation methods are investigated, including: domain-specific SMT tuning, mixture modeling, and the so called provenance features. Applying the developed methods on a competitive dialectal Arabic to English translation task, where the Arabic data contains a mixture of dialects, our results show that using the classifier output improves over the meta-information based splits. We also show that some adaptation methods can hurt the performance, and a combination of techniques is required to guarantee improvements. Finally, we perform simple system selection of the dialect-specific SMT systems and show that we can achieve gains for all dialects.

The paper is structured as follows. We review related work in Section 2. The automatic dialect classifier is introduced in Section 3 and the adaptation methods in Section 4. The experimental setup is described in Section 5. Classification and translation results along with an analysis are discussed in Section 6 and Section 7 correspondingly. Lastly, we conclude with few suggestions for future work in Section 8.

2 Related Work

Various adaptation techniques have been suggested in the past for SMT. The techniques use either meta-information to define the different corpora, e.g., (Foster and Kuhn, 2007; Chiang et al., 2011) or automatic clustering methods, e.g., (Eidelman et al., 2012; Sennrich, 2012a), and focus on training data splitting. We differ from previous work by using automatic dialect classification to refine the splitting of the training data. Furthermore, we use the classifier to split the tuning and test sets, build dialect specific systems and combine them using system selection based on the dialect classification.

Interest in techniques for handling varieties of a language has been growing in the last few years. In 2014, two workshops will be held dealing with resources, techniques and tools specialized for language varieties, LT4CloseLang[1] at EMNLP and VarDial[2] at COLING. The discriminating similar languages (DSL) shared task (Tan et al., 2014) offers an opportunity for consistent comparison of different classification methods. The DSL evaluation is done mainly on European languages. In this work, we focus on dialectal varieties of the Arabic language. Nevertheless, the methods developed are generic and can be applied to other languages. Zbib et al. (2012) discuss machine translation of Arabic dialects. Using human annotated dialectal data, they achieve improvements over a general SMT system. We differ from their work by using automatic dialect classification for SMT. Previous work on dialect classification discussed

[1] http://www.c-phil.uni-hamburg.de/view/Main/LTforCloseLang2014

[2] http://corporavm.uni-koeln.de/vardial/

the definition of the problem, and built automatic classifiers (Zaidan and Callison-Burch, 2011; Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2014). Ideas for applying the classifier for SMT were discussed but not implemented. In this work, we implement a competitive dialect classifier which is then successfully applied for SMT and shows strong improvements over a competitive baseline. To the best of our knowledge, this work presents the first successful application of automatic dialect classification for SMT.

3 Dialect Classification

The task of dialect classification attempts to identify the dialect of a given sentence. In this work, we use a supervised sentence-level dialect classifier to designate sentences as MSA or ARZ. Sentences with ARZ content or structure are identified as ARZ, otherwise they are marked as MSA (Zaidan and Callison-Burch, 2014). The classifier is trained on the **Arabic Online Commentary Set** (AOC) (Zaidan and Callison-Burch, 2011). The data consists of commentary by online readers of Arabic newspapers with a high degree of dialectical content, together with human-annotated labels indicating the dialect of each sentence. The data had been obtained by a crowd-sourcing effort. In the current paper, we focus on the MSA-ARZ [3] split of the data. The split contains 25K sentences and 650K words, where around half of the sentences are annotated as MSA and half as ARZ.

To implement the automatic sentence classifier, we use a linear SVM framework, i.e., the open source *LIBLINEAR* toolkit (Hsieh et al., 2008; Fan et al., 2008). The trainer can easily handle a large number of instances and features. As the objective function, we use $L1$ regularized $L2$-loss support vector classification[4]. We set the penalty term $C = 0.5$. To classify a sentence $t_1^n = t_1 ... t_n$, we compute a linear score $s(t_1^n)$ as follows:

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

where $\phi_s(c_i, t_i)$ is a binary feature function which takes into account the context c_i of token t_i. The weight vector $\mathbf{w} \in \mathbb{R}^d$ is a high-dimensional vector obtained during training. In our experiments, we classify a tokenized sentence as being Egyptian Dialectal (ARZ) if $s(t_1^n) > 0$.

The feature functions we use include token (word-level) unigram and bigram, Part-of-Speech unigram and dictionary based features. The features are combined according to Eq. 1. We described the used classifier and features in more detail in (Tillmann et al., 2014).

4 Adaptation Methods

In this section, we introduce different approaches to domain adaptation that will be utilized to generate dialect-specific SMT systems.

[3] Note that we denote Egyptian Arabic with ARZ instead of the EGY label used by (Zaidan and Callison-Burch, 2011). ARZ is the standard ISO language code for Egyptian Arabic.

[4] In the LIBLINEAR toolkit settings, we use solver type 5 with default termination criterion.

4.1 SMT tuning

We carry out a weight vector $\lambda_1^M = \lambda_1 ... \lambda_M$ tuning of a standard log-linear SMT model:

$$q(e_1^I, f_1^J) = \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J), \tag{2}$$

where (e_1^I, f_1^J) are target and source sentences of length (I, J) correspondingly, and h_1^M are feature functions.

We use pairwise ranking optimization (PRO) (Hopkins and May, 2011) to tune the scaling factors. Instead of optimizing the log-linear model probability, PRO directly optimizes the final translation quality. To perform adaptation using PRO tuning, the development set can be varied to represent different domains. We experiment with using different development sets obtained from different domains, as well as using the dialect classifier to obtain dialect specific development sets. The scaling factors obtained for a specific tuning set will represent an adapted system for the domain of the tuning set.

4.2 Mixture tuning

Mixture modeling is a technique for combining several models using weights assigned to the different components. Domain adaptation could be achieved using mixture modeling when the weights are related to the proximity of the components to the domain being translated. As we generate several translation models differing by the training corpora domain, interpolating these models could yield further improvements. In this work, we focus on mixture modeling using linear interpolation.

Linear interpolation is a commonly used framework for combining different SMT models (Foster and Kuhn, 2007). Given n phrase models $p_1^n = p_1 ... p_n$, and λ_1^n interpolation weights, linear interpolation is defined as follows:

$$p(\tilde{f}|\tilde{e}; \lambda) = \sum_i \lambda_i \cdot p_i(\tilde{f}|\tilde{e}) \tag{3}$$

In this work, the interpolation weights are optimized over a development set which represents a specific domain. We use the phrase model perplexity as an objective function:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \left\{ - \sum_{(\tilde{f},\tilde{e})} \frac{1}{N} \log p(\tilde{f}|\tilde{e}; \lambda) \right\} \tag{4}$$

$(\tilde{f}, \tilde{e})$ are phrase pairs extracted from the development set using standard phrase extraction methods (symmetrized word alignment and heuristic phrase extraction). We use the L-BFGS optimization technique as done by (Sennrich, 2012b). Note that we apply linear interpolation to all extracted rules (including phrase, hierarchical, and tree-to-string rules).

4.3 Provenance features

Chiang et al. (2011) suggest provenance features for improving SMT performance. Instead of

	Corpus	#Sent	#Ar tok	Meta-info. dialect	Automatic dialect classification			
					ARZ(%)		MSA(%)	
Train	Forums	299K	4.1M	ARZ	183K	(61%)	116K	(39%)
	Broadcast	169K	3.9M	MSA	18K	(11%)	151K	(89%)
	Newswire	885K	24.9M	MSA	29K	(3%)	856K	(97%)
	Other	726K	5M	MIX	184K	(25%)	542K	(75%)
	All	2.1M	37.9M	MIX	415K	(20%)	1 663K	(80%)
Tune	DEV12	1 209	17 702	ARZ	507	(42%)	702	(58%)
	P1R6	2 715	52 206	ARZ	1481	(55%)	1 234	(45%)
	DEV10-wb	968	42 092	MSA	49	(5%)	919	(95%)
Dev	DEV12	1 510	27 134	ARZ	584	(39%)	926	(61%)
	P1R6	1 137	17 991	ARZ	739	(65%)	398	(35%)
	DEV10-wb	1 059	42 563	MSA	46	(4%)	1 013	(96%)

Table 1: BOLT parallel training data by genre, meta-information dialect and automatic dialect classification results. MIX may contain additional Arabic dialects. The number of sentences (#Sent) and Arabic tokens (#Ar Tok) are given. We report the percentage of sentences classified as ARZ or MSA for each of the corpora listed.

training one model on the whole data, they suggest to condition the models on the provenance, i.e., the meta-information (genre, collection) of the corpus the data is coming from.

In this work, we use IBM Model 1 lexical smoothing as provenance features. Splitting the training data into n sub-corpora z_1^n, we introduce $2 \cdot n$ provenance features (for standard and inverse Model 1 directions) into the log-linear framework of SMT. - The log-linear weights of the provenance features are optimized as part of the PRO tuning of the whole set of SMT features. Adaptation is then achieved by tuning the weights of the features to improve performance on a target dialect tuning set.

5 Experimental Setup

5.1 Training corpora

We evaluate our dialect adaptation methods empirically in the context of the BOLT Phase 2 Dialectal-Arabic-to-English task[5]. The dialect chosen for Phase 2 is Egyptian Arabic (ARZ). The BOLT program goes beyond previous projects, shifting the focus from translating structured standardized text, such as Modern Standard Arabic (MSA) newswire, to a user generated noisy text such as Arabic dialect forums or sms. Translating Arabic dialects is a challenging task due to the scarcity of training data and the lack of common orthography causing a larger vocabulary size and higher ambiguity. Due to the scarcity of the ARZ training data, MSA resources are being utilized for the project. In such a scenario, an important research question arises on how to use the MSA data in the most beneficial way to translate the given dialect.

The training data for the BOLT Phase 2 program is summarized in Table 1. The table includes information about domain, dialect and size (automatic classification results are discussed in Section 6.1). Preprocessing includes Arabic tokenization and segmentation based on

[5] `http://www.nist.gov/itl/iad/mig/upload/BOLT_phase2_MT_evalplan_v8.pdf`

(Lee et al., 2003). English preprocessing includes lowercasing and punctuation tokenization.

The Forums data was collected from Egyptian webforums, therefore it is written mainly in the ARZ dialect. Broadcast data was collected and transcribed manually from various Arabic TV sources. In the broadcast domain speakers usually use MSA but sometimes also switch (for a short phrase) to dialectal speech. The newswire data is mostly written in MSA. We tune and test the SMT systems using 3 sets: DEV12 is extracted from LDC2012E30-BOLT Phase 1 DevTest Source and Translation V4, and has 1 reference, P1R6 from LDC2012E124-BOLT Phase 1 Translation Training Data R6 (1 reference), and DEV10-wb from LDC2010E30 GALE Phase 5 DevTest NW & WB Translations V3.0 (4 references). Note that the sets include two parts, a tune part which is used mainly for PRO tuning and a held-out development part which is used for testing and will be displayed in the results. Most of the BOLT training data is available through the linguistic data consortium (LDC) and is regularly part of the NIST open MT evaluation [6]. For language model training purposes, we use an additional 8 billion words (4B words from the LDC gigaword corpus and 4B words collected from web resources).

5.2 Translation system

We use an in-house implementation of a chart-based decoder (Zhao and Al-Onaizan, 2008). The decoder utilizes phrase, hierarchical, and tree-to-string rules to perform derivations. For the tree-to-string grammar, the source side of the parallel training data is parsed and word-alignment is performed. Tree-to-string rules together with their probabilities are then automatically learned from the data (Liu et al., 2006). Reordering patterns can be learned from linguistic labels assigned to chunks by combining parsing and alignment information. For Example, the rule [X,VP][X,VB][X,NP] → [X,NP][X,VB] rewrites a VP with two constituents VB and NP into an NP VB order in the target. The tree-to-string grammar bounds the search space to the available reordering patterns. However, if the correct word order cannot be generated by the tree-to-string grammar, the system resorts to hierarchical or phrase based rules to extend the coverage.

The hypothesis score is defined by the standard log-linear model combination, which includes in this case count-based features for phrase, glue, hierarchical and tree-to-string rules. Additional standard models such as length penalty and lexical smoothing are also incorporated into the decoder. All MT experiments are optimized with PRO to minimize the combined error measure of BLEU (Papineni et al., 2002) and TER (Snover et al., 2006), (TER-BLEU)/2.

6 Classification Results

In this section, we present classification accuracies as well as classification results on the SMT training data. We train a dialect classifier as suggested in Section 3. The classifier performance is presented in Table 2. The table includes two sets of experiments, a 10-fold cross validation using the MSA-ARZ portion of the Arabic Online Commentary (AOC) data, and the performance on DEV12 (tune part in Table 1) when training the classifier on the whole AOC data. The performance is measured in terms of accuracy (fraction of sentences correctly tagged) and dialect precision and recall, e.g., for ARZ:

[6] For a list of the NIST MT12 corpora, see http://www.nist.gov/itl/iad/mig/upload/OpenMT12_LDCAgreement.pdf

		ARZ			MSA		
Set	Acc	P	R	F	P	R	F
cross-validation	89.1	91.1	85.7	88.3	87.5	92.2	89.8
DEV12	87.8	92.8	83.0	87.6	83.4	93.0	88.0

Table 2: Arabic dialect classification results: predicting MSA vs. ARZ. Accuracy (Acc), precision (P), recall (R) and F-measure (F) are given in percentages [%].

$$\textbf{precision} = \frac{\text{\# ARZ correctly tagged}}{\text{\# ARZ tagged}}, \quad \textbf{recall} = \frac{\text{\# ARZ correctly tagged}}{\text{\# ARZ reference}}$$

The classifier achieves high precision results on the ARZ portion of the sets. This is important for the adaptation experiments as we are mostly interested in adapting the systems towards the ARZ dialect. As most of the training data is MSA, having a correctly classified portion of ARZ data will help gain more improvements when adapting towards the ARZ domain.

In comparison to state-of-the-art, Elfardy and Diab (2013) report 85.3% accuracy for their best setup on a similar 10-fold cross-validation experiment. Zaidan and Callison-Burch (2014) report 87.9% accuracy on a similar setup. Our results show an improvement of 1.2% absolute over the best reported results on this task.

6.1 Classifier analysis

In this section, we run the classifier over the BOLT data and measure its dialectal degree, and whether the dialectal degree corresponds to the labeled provenance of the data. Classification statistics are presented in Table 1, where we report the number and percentage of sentences classified as ARZ or MSA. The ARZ 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. We conclude that the data contains a mixture of dialects, and a more refined splitting using the dialect classification information could help improve adaptation methods.

Classifications examples from the BOLT data are given in Table 3. In the first document fragment, the user starts with MSA sentences, then switches to ARZ 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 these sentences correctly. The second text fragment shows 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 corresponding binary based feature 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 fire and features that correspond to Arabic prefixes tend to support a classification as ARZ.

Classification result		Arabic	English
correct	MSA	انا قرات الموضوع والردود .	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
incorrect	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 ca n't remember what he told me

Table 3: Automatic classification examples. The classes ARZ and MSA, Arabic source and English target sentences are given. Dialectal words are in **bold**.

Tuning set	DEV12 (T-B)/2	P1R6 (T-B)/2	DEV10 (T-B)/2
baseline	15.35	15.02	0.73
TUNE.MSA	15.52	15.15	0.73
TUNE.ARZ	15.58	15.05	1.15

Table 4: PRO tuning adaptation: The baseline is tuned using P1R6+DEV10, the TUNE.MSA and TUNE.ARZ sets are based on the classifier output over the concatenation of all tuning sets.

7 Translation Results

The baseline SMT system used in this work is based upon a tree-to-string decoder as described in Section 5.2. To create the rule tables, we use the concatenation of three word alignments, namely, HMM, IBM model 4 and maximum entropy aligner to maximize performance (Tu et al., 2012). The PRO tuning is done using the concatenation (P1R6+DEV10-wb).tune, as it performed best among all possible combinations.

Next, we experiment with the various adaptation methods suggested in Section 4. We focus on the comparison between using splits based on the meta-information and splits based on the automatic classifier output (Table 1).

7.1 SMT tuning

Tuning an SMT system using a domain-specific tuning set can adapt the scaling factors towards the target domain. For example, a bigger word-penalty scaling factor will encourage shorter sentences. Using meta-information based tuning sets, we found that the best combination is to use P1R6+DEV10-wb for tuning. To experiment with tuning sets based on dialect classification, we concatenate all tuning sets into TUNE=DEV12+P1R6+DEV10-wb. We then split the concatenated tuning set into an ARZ and an MSA part based on the classifier output and denote these splits TUNE.ARZ and TUNE.MSA respectively.

A comparison between the baseline system and the tuning using the classifier-based splits

Tune set		Forum	Broadcast	Newswire	Other	All
Meta-info.	P1R6+DEV10	0.23	0.26	0.34	0.06	0.12
	DEV12+P1R6	0.40	0.15	0.05	0.03	0.37
Classifier	TUNE.ARZ	0.62	0.11	0.03	0.06	0.17
	TUNE.MSA	0.09	0.23	0.19	0.03	0.46

Table 5: Optimal mixture weights for different tuning sets. The tuning sets are constructed using meta-information or the classifier output.

over the dev sets is given in Table 4. From the results, we note that tuning towards MSA does not hurt the results on DEV10-wb-dev, with a slight degradation on the ARZ dev sets. Tuning towards ARZ degrades the results on the ARZ dev sets and a bigger degradation is observed on the DEV10 set. Examining the scaling factors for the ARZ tuning set, almost no change is observed (in comparison to the baseline). We hypothesize that without domain-specific models, tuning towards a target domain is not effective. In the following experiments, we introduce more adaptation into the system and re-apply the PRO adaptation.

7.2 Mixture tuning

In this section, we experiment with optimizing the linear mixture weights towards a specific dialect as presented in Section 4.2. The optimization is done using phrase perplexity as the objective function. As components for the mixture, we use the meta-information split from Table 1. As tuning sets, we evaluate the performance of the meta-information based sets versus using the classifier. Note that we do not split the training corpora further according to dialects here, but concentrate on the tuning of the mixture weights instead.

The resulting optimal weights from the L-BFGS optimization are presented in Table 5. Note that we use *All* data (a concatenation of all corpora) as an additional corpus to ensure optimal translation results. The first block of weights is based on meta-information tuning sets. We experiment with the baseline SMT system tuning set as-well-as a supposedly ARZ tuning set (DEV12+P1R6). P1R6+DEV10-wb contains mostly MSA sentences, therefore the Newswire corpus is assigned the highest weight. When using the DEV12+P1R6 tuning set, the weight shifts to the Forums and All data, as they are the corpora most similar to the tuning set.

For the classifier split tuning sets, when tuning on TUNE.ARZ, weight is shifted to the Forums based model. Tuning on the MSA part TUNE.MSA, the weight shifts back to Broadcast, Newswire and All data, which contain a majority of MSA sentences. To summarize, we note that mixture tuning is producing expected results, and using the classifier splits assigns higher weights to the corresponding dialect. Therefore, the classifier based mixture tuning is more reliable and generates better contrast between the corpora. Next, we use the weights based on the classifier splits to create interpolated rule tables and build SMT systems using those tables.

The SMT results of different mixture modeling experiments are summarized in Table 6. Performing linear interpolation of the rule tables using uniform weights (linear.uniform) already achieves gains over the baseline. Note that PRO retuning using the baseline tuning set (P1R6+DEV10) is performed, unless stated otherwise. Linear interpolation with weights optimized on the MSA set (the weights associated with TUNE.MSA in Table 5), linear.MSA, achieves further gains on the MSA DEV10-wb-dev set, with a loss on the ARZ sets as expected.

System	DEV12 (T-B)/2	P1R6 (T-B)/2	DEV10 (T-B)/2
baseline	15.35	15.02	0.73
linear.uniform	15.17	14.89	0.59
linear.MSA	15.36	15.05	0.33
+pro.MSA	15.35	15.31	0.13
linear.ARZ	15.04	14.60	1.14
+pro.ARZ	15.22	14.67	1.41
system selection	15.22	14.67	0.15

Table 6: Mixture tuning adaptation: linear interpolation using uniform weights (linear.uniform), ARZ and MSA optimized weights are compared. +pro.Dialect indicates PRO tuning adaptation.

Performing PRO tuning using TUNE.MSA over the linear.MSA system achieves further gains on the MSA set. The total gain over the baseline is -0.6% (T-B)/2 for DEV10-wb-dev. The resulting system is adapted for MSA sentences and performs well under this condition.

To build a system targeting the ARZ dialect, we repeat the same procedure as for MSA. We start with linear interpolation using ARZ optimized mixture weights (linear.ARZ), which achieves -0.3% and -0.4% improvements over DEV12-dev and P1R6-dev correspondingly. Loss is observed on the MSA set as expected. Adding the PRO tuning using the ARZ classified sentences hurts the results in this case, with a 0.2% degradation on DEV12-dev.

Finally we experiment with selecting hypotheses from the output of the MSA optimized system (linear.MSA+pro.MSA) and the ARZ optimized one (linear.ARZ+pro.ARZ). The idea is to use the MSA optimized system for MSA classified sentences and the ARZ optimized system for ARZ sentences. Using this selection, we might see improvements on the whole dev sets, as it might be the case that the ARZ system improved on the ARZ sentences and got much worse on the MSA sentences, masking the gains on the whole dev set. The system selection retains the gains on the MSA data, but not on the ARZ sets. We conclude that the ARZ system in this case did not improve on the ARZ part of the data. Next, we add domain specific models to the SMT system, giving more flexibility for PRO to overweight dialect specific features and target the ARZ dialect.

7.3 Provenance features

To compare meta-information based and classifier based corpora splits for provenance features, we devise two provenance setups: 1) m1Manual, manual splitting with 4 corpora, Forums, Broadcast, Newswire and Other to train Model 1 models in standard and inverse directions (8 additional features in the decoder), 2) m1Class, classifier based splitting, Forums, Broadcast and Other corpora are split into MSA and ARZ parts using the classification, the Newswire corpus is kept intact as it is mostly MSA (14 additional features in the decoder).

The results using the provenance features on top of the dialectal optimized systems are given in Table 7. From the results, we note that adding provenance features achieves further improvements, and using m1Class has a slight edge over the m1Manual provenance features. In this case, the system selection (from (lin+pro).MSA+m1Class and (lin+pro).ARZ+m1Class)

System	DEV12 (T-B)/2	P1R6 (T-B)/2	DEV10 (T-B)/2
baseline	15.35	15.02	0.73
(lin+pro).MSA	15.35	15.31	0.13
+m1Manual	15.39	15.10	0.21
+m1Class	15.35	15.05	0.19
(lin+pro).ARZ	15.22	14.67	1.41
+m1Manual	15.46	14.78	1.94
+m1Class	15.41	14.61	1.98
System selection			
m1Class	15.07	14.52	0.31

Table 7: Adaptation based on provenance features: systems with provenance features derived from manual splits (m1Manual) and classifier-based splits (m1Class) are compared.

System	DEV12 (T-B)/2	P1R6 (T-B)/2
	MSA subset	
(lin+pro).MSA	14.33	12.03
+m1Manual	14.41	11.65
+m1Class	14.39	11.59
(lin+pro).ARZ	14.32	12.21
	ARZ subset	
(lin+pro).ARZ	16.79	15.84
+m1Manual	16.60	15.98
+m1Class	16.29	15.82
(lin+pro).MSA	17.16	16.77

Table 8: Provenance adaptation for the MSA and ARZ subsets of the dev sets. The dev sets are split using the automatic classification.

is performing well, combining the best of both systems. Comparing the baseline to the system selection, we achieve 0.3% improvement over DEV12, 0.5% over P1R6 and 0.4% over DEV10.

To analyze the results further, we split the DEV12 and P1R6 dev sets into the corresponding dialectal parts, and measure the effect of adding provenance features over these parts. In such a case, we expect that ARZ optimized systems will improve over the ARZ part, while MSA optimized systems will improve over the MSA part. The results are summarized in Table 8. Note that in this table we are using subsets of the dev sets. Concentrating on the MSA part of the dev sets, we note that adding the provenance features is improving mainly on P1R6, with a slight gain for classifier based provenance (m1Class) over meta-information based (m1Manual). As a contrast, the ARZ optimized system is performing poorly on the MSA parts of the dev sets. The picture is similar for the ARZ part of the dev sets, this time the main improvement is on the DEV12 set, with a bigger gain for m1Class over m1Manual, 0.3% (T-B)/2.

7.4 Translation examples

In this section, we perform manual translation error analysis. Translation examples are given in Table 9. The examples show that the system selection of dialectal optimized systems (sel.) improves over the baseline (base). The first two examples are ARZ sentences while the last is an MSA one. These examples were selected to demonstrate the difficulty of dialectal language translation and to show how a dialect classifier can remedy the problems encountered.

In the first sentence, the word العربية means ‘Arab’ but only in ARZ it could also mean ‘car’. The sentence is classified correctly, and the ARZ optimized system is able to generate the correct lexical meaning of the word. Similarly, in the second example, the word بقى means ‘has become’ in MSA and ‘is’ in ARZ. The ARZ system generates a better translation. The third sentence is an MSA sentence, where the baseline has a reordering error of ‘controls’ being generated before ‘professional’, and the word تراعي (consider) is dropped. The MSA optimized system generates a better reordering as-well-as a better lexical choice. We conclude

src	كدة جبتي سيرة **العربية**
ref	you mentioned the **car**
base	this way you brought the biography of the **arab**
sel.	why did you bring the biography of the **car**
src	ده **بقى** عقاب اشد و اقوى
ref	that punishment **is** harder and tougher
base	this **has become** stronger and stronger punishment
sel.	this punishment **is** harder and stronger
src	دون ان تراعي **الضوابط** المهنية
ref	without taking into account professional **standards**
base	without that **controls** that professional
sel.	without that they consider professional **rules**

Table 9: Sample sentences. The source, reference, baseline hypothesis and system selection (sel.) hypothesis are given.

that ignoring the effects of dialectal data in MT makes the task even more ambiguous, and dialectal identification is crucial to lessen the ambiguity and improve the lexical choice.

8 Conclusions and Future Work

In this work, we implement and successfully apply an automatic dialect classifier for SMT. The classifier is applied on the BOLT task, where we compare meta-information based data splits versus using the classifier output. The various splits are utilized for three adaptation methods: PRO tuning adaptation, mixture adaptation and provenance features. For mixture adaptation, our results show that the classifier based splits generate better contrast between the different training corpora weights, where more emphasis is placed on the ARZ forums data when using the ARZ tuning set based on the classifier output compared to the ARZ tuning set based on meta-information. For PRO tuning adaptation, we conclude that using the classifier splits without additional dialect specific models is not helpful and can degrade the performance. When adding the provenance features, a system selection of ARZ and MSA optimized systems improves over the baseline by 0.5% on the ARZ dev set.

In future work, it would be interesting to measure the effect of the classifier quality for the adapted SMT systems. For mixture modeling, we started experimenting with training data splitting by the classifier to create dialect specific rule tables and perform rule table interpolation. A problem occurs when optimizing the mixture weights, where some of the ARZ splits were assigned lower weights than the MSA counterparts when optimizing towards ARZ. We hypothesize that this result is obtained due to many unknown phrase pairs in the ARZ tables which are rather small in size. Smoothing for unknown phrase pairs should be applied when more splits are used and sparseness becomes a problem. Many other techniques for adaptation using dialect classification could be experimented with in future work. For example, phrase level classification, or using the classifier scores as a feature in the SMT decoder.

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

Chiang, D., DeNeefe, S., and Pust, M. (2011). Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 455–460, Portland, Oregon, USA. Association for Computational Linguistics.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

Elfardy, H. and Diab, M. (2013). Sentence Level Dialect Identification in Arabic. In *Proc. of the ACL 2013 (Volume 2: Short Papers)*, pages 456–461, Sofia, Bulgaria. Association for Computational Linguistics.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: a Library for Large Linear Classification. *Machine Learning Journal*, 9:1871–1874.

Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic. Association for Computational Linguistics.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proc. of the EMNLP 2011*, pages 1352–1362, Edinburgh, Scotland, UK.

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

Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., and Hassan, H. (2003). Language Model Based Arabic Word Segmentation. In Hinrichs, E. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 399–406.

Liu, Y. ., Liu, Q., and Lin, S. (2006). Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Sennrich, R. (2012a). Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *Proceedings of the 16th EAMT Conference*, pages 185–192, Trento, Italy.

Sennrich, R. (2012b). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France. Association for Computational Linguistics.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Tan, L., Zampieri, M., Ljubešic, N., and Tiedemann, J. (2014). Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *7th Workshop on Building and Using Comparable Corpora*.

Tillmann, C., Mansour, S., and Yaser, A.-O. (2014). Improved sentence-level arabic dialect classification. In *COLING Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 110–119, Dublin, Ireland.

Tu, Z., Liu, Y., He, Y., van Genabith, J., Liu, Q., and Lin, S. (2012). Combining multiple alignments to improve machine translation. In *Proceedings of COLING 2012: Posters*, pages 1249–1260, Mumbai, India. The COLING 2012 Organizing Committee.

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proc. of ACL / HLT 11*, pages 1220–1229, Portland, Oregon, USA. Association for Computational Linguistics.

Zaidan, O. F. and Callison-Burch, C. (2014). Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Zbib, R., Malchiodi, E., Devlin, J., Stallard, D., Matsoukas, S., Schwartz, R. M., Makhoul, J., Zaidan, O., and Callison-Burch, C. (2012). Machine Translation of Arabic Dialects. In *HLT-NAACL*, pages 49–59.

Zhao, B. and Al-Onaizan, Y. (2008). Generalizing local and non-local word-reordering patterns for syntax-based machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 572–581, Honolulu, Hawaii. Association for Computational Linguistics.