Abstract

We report the results of our work on automating the transliteration decision of named entities for English to Arabic machine translation. We construct a classification-based framework to automate this decision, evaluate our classifier both in the limited news and the diverse Wikipedia domains, and achieve promising accuracy. Moreover, we demonstrate a reduction of translation error and an improvement in the performance of an English-to-Arabic machine translation system.

1 Introduction

Translation of named entities (NEs) is important for NLP applications such as Machine Translation (MT) and Cross-lingual Information Retrieval. For MT, NEs are a major subset of the out-of-vocabulary terms (OOVs). Due to their diversity, they cannot always be found in parallel corpora, dictionaries or gazetteers. Thus, state-of-the-art of MT needs to handle NEs in specific ways. For instance, in the English-Arabic automatic translation example given in Figure 1, the noun "North" has been erroneously translated to "الشماليّة /AlSmAlyp " (indicating the north direction in English) instead of being transliterated to "ثورة / nwrv".

As shown in Figure 1, direct translation of in-vocabulary terms could degrade translation quality. Also blind transliteration of OOVs does not necessarily contribute to translation adequacy and may actually create noisy contexts for the language model and the decoder.

Figure 1: Example of a NE translation error.

An intelligent decision between translation and transliteration should use semantic and contextual information such as the type of the named-entity and the surrounding terms. In this paper, we construct and evaluate a classification-based framework to automate the translation vs. transliteration decision. We evaluate our classifier both in the limited news and diverse Wikipedia domains, and achieve promising accuracy. Moreover, we conduct an extrinsic evaluation of the classifier within an English to Arabic MT system. In an in-domain (news) MT task, the classifier contributes to a modest (yet significant) improvement in MT quality. Moreover, for a Wikipedia translation task, we demonstrate that our classifier can reduce the erroneous translation of 60.5% of the named entities.

In summary our contributions are: (a) We automatically construct a bilingual lexicon of NEs paired with the transliteration/translation decisions in two domains.1 (b) We build a binary classifier for transliteration and translation decision with a promising accuracy (c) We demonstrate its utility

1The dataset can be found at http://www.qatar.cmu.edu/~behrang/NETLexicon.
within an MT framework.

2 Learning when to transliterate

We model the decision as a binary classification at the token level. A token (within a named-entity) gets translation or transliteration label. In "Dudley North" and "North London", our classifier is expected to choose transliteration of "North" in the former case, as opposed to translation in the latter. The binary decision needs to use a rich set of local and contextual features. We use the Support Vector Machines as a robust framework for binary classification using a set of interdependent features.

2 We build two classifiers: (a) Classifier $C_{\text{news}}$, trained on a large set of distinct NEs extracted from news-related parallel corpora; and (b) Classifier $C_{\text{diverse}}$, trained on a combination of the news related NEs and a smaller set of diverse-topic NEs extracted from Wikipedia titles. We evaluate the two classifiers in both news and the diverse domains to observe the effects of noise and domain change.

2.1 Preparing the labeled data

Our classifier requires a set of NEs with token-level gold labels. We compile such data from two resources: We heuristically extract and label parallel NEs from a large word aligned parallel corpus and we use a lexicon of bilingual NEs collected from Arabic and Wikipedia titles. Starting with a word aligned parallel corpus, we use the UIUC NE tagger (Ratinov and Roth, 2009) to tag the English sentences with four classes of NEs: Person (PER), Location (LOC), Organization (ORG) and Miscellaneous (MISC). Furthermore, we use the word alignments to project and collect the span of the associated Arabic named-entities. To reduce the noisy nature of word alignments, we designed a procedure to clean up the noisy Arabic NE spans by POS verification, and heuristically filtering impossible items (e.g. verbs). This results in a bilingual lexicon of about 57K named-entity pairs. The distribution of NEs categories is reported in Table 1.

To train and evaluate the $C_{\text{diverse}}$ classifier, we expand our labeled data with Wikipedia NEs using the cross-lingual hyperlinks. Wikipedia article titles often correspond to NEs (Kazama and Torisawa, 2007) and have been already used in different works for NEs recognition (Nothman et al., 2013) and disambiguation (Cucerzan, 2007). We improve the Arabic-English Wikipedia title lexicon of Mohit et al. (2012) and build a Wikipedia exclusive lexicon with 4K bilingual entities. In order to test the domain effects, our lexicon includes only NEs which are not present in the parallel corpus. The statistics given in Table 1 demonstrate different nature of the labeled datasets. The two datasets were labeled semi-automatically using the transliteration similarity measure ($Fr$) proposed by Freeman et al. (2006), a variant of edit distance measuring the similarity between an English word and its Arabic transliteration. In our experiments, English tokens having an $Fr > 0.6$ are considered as transliteration, others having $Fr < 0.5$ as translation. These thresholds were determined after tuning with a held out development set. For tokens having $Fr$ between 0.5 and 0.6, the decision is not obvious. To label these instances (around 5K unique tokens), we manually transliterate them using Microsoft Maren tool. We again compute the $Fr$ between the obtained transliteration, in its Buckwalter form and the corresponding English token and use the same threshold to distinguish between the two classes. Some examples of NEs and their appropriate classes are presented in Table 2.

2.2 Classification Features

We use a total of 32 features selected from the following classes:

Token-based features: These consist of several features based on the token string and indicate

<table>
<thead>
<tr>
<th>Transliteration</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minnesota ↔ مينيسوتا</td>
<td>Agency ↔ الوكالة</td>
</tr>
<tr>
<td>Fluke ↔ فلاك</td>
<td>Islamic ↔ الإسلامية</td>
</tr>
</tbody>
</table>

Table 2: Examples of NEs labeled using Freeman Score.

---

2 We use the LIBSVM package (Chang and Lin, 2011).

whether the token is capital initial, composed entirely of capital letters, ends with a period (such as Mr.), contains a digit or a Latin number (e.g. Muhammad II) or contains punctuation marks. The string of the token is also added as a feature. We also check if the token is a regular noun in the WORDNET (Fellbaum, 1998) which increases its chance of being translated as opposed to transliterated.

Semantic features: These features mainly indicate the NE category obtained using an NE tagger. We also define a number of markers of person (such as Doctor, Engineer, etc.) and organization (such as Corp.) names. We used the list of markers available at: http://drupal.org/node/1439292, that we extended manually.

Contextual features: These features are related to the token’s local context within the NE. These include information about the current token’s surrounding tokens, its relative position in the NE (beginning, middle or end). Another feature represents the length of the NE in number of tokens.

2.3 Experiments

We train two classifiers and tune their parameters using a held out development set of 500 NEs drawn randomly from the news parallel corpus. We use 55k NEs from the same corpus to train the C_news classifier. Furthermore, we train the C_diverse classifier cumulatively with the 55K news NEs and another 4600 NEs from Wikipedia titles.

The classifiers are evaluated on three different datasets: Test_News which consists of 2K of NEs selected randomly from the news corpus, Test_Wiki consisting of 1K NEs extracted from the Wikipedia and Test_Combination, an aggregation of the two previous sets. We manually reviewed the labels of these test sets and fixed any incorrect labels. Table 3 compares the accuracy of the two classifiers under different training and test data settings. Starting with a majority class baseline, our classifiers achieve a promising performance in most settings. The majority class for both classifiers is the translation which performs as a baseline approach with an accuracy equal to the distribution of the two classes. We also observe that the addition of a small diverse training set in C_diverse provides a relatively large improvement (about 2%) when tested on Wikipedia. Finally, Figure 2 illustrates the contribution of different classes of features on our diverse classifier (evaluated on Test_Wiki). We observe a fairly linear relationship between the size of the training data and the accuracy. Furthermore, we observe that the features describing the category of the NE are more important than the token’s local context. For example, in the case of “Dudley North” and ”North London”, the most effective feature for the decision is the category of the named entities.

<table>
<thead>
<tr>
<th></th>
<th>Test_News</th>
<th>Test_Wiki</th>
<th>Test_Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>56.70</td>
<td>57.09</td>
<td>56.89</td>
</tr>
<tr>
<td>C_news</td>
<td>90.40</td>
<td>84.10</td>
<td>88.64</td>
</tr>
<tr>
<td>C_diverse</td>
<td>90.42</td>
<td>86.00</td>
<td>89.18</td>
</tr>
</tbody>
</table>

Table 3: Accuracy results for the two classifiers and the baseline on the three test datasets.

3 Extrinsic MT evaluation

We evaluate the effects of the classifier on an English to Arabic statistical MT system. Our first evaluation focuses on the utility of our classifier in preventing erroneous translation of NEs which need to be transliterated. In the following experiments we use C_news classifier. In order to experiment with a diverse set of NEs, we conducted a study on a small corpus (98,197 terms) of Wikipedia articles from a
diverse set of topics. We use 10 Wikipedia articles describing: Anarchism, Artemis, Buddhism, Isfahan, Shawn Michaels, Turkey, etc. We first use our classifier to locate the subset of NEs which should be transliterated. An annotator validates the decision and examines the phrase table on the default MT decision on those NEs. We observe that out of 1031 NE tokens, 624 tokens (60.5%) which would have been translated incorrectly, are directed to the transliteration module.

Finally, we deploy the transliteration classifier as a pre-translation component to the MT system. Our MT test set is the MEDAR corpus (Maegaard et al., 2010). The MEDAR corpus consists of about 10,000 words English texts on news related to the climate change with four Arabic reference translations. Due to the lack of non-news English-Arabic corpus, we have to limit this experiment only to the news domain. However, we expect that many of the NEs may already exist in the training corpus and the effects of the classifier is more limited than using a diverse domain like Wikipedia. We automatically locate the NEs in the source language sentences and use the classifier to find those which should be transliterated. For such terms, we offer the transliterated form as an option to the decoder aiming to improve the decoding process. For that a human annotator selected the transliterations from the suggested list that is provided by the automatic transliterator (Maren) without any knowledge of the reference transliterations.

Table 4 shows the impact of adding the classifier to the SMT pipeline with a modest improvement. Moreover, a bilingual annotator examined the automatically tagged NEs in the MT test set and labeled them with the translation vs. transliteration decisions. Having such gold standard decisions, we evaluated the classifier against the MT test set. The classifier’s accuracy was 89% which is as strong as the earlier intrinsic evaluation. The false positives are 5% which represents around 12.6% of the total errors.

The following example shows how our classifier prevents the MT to choose a wrong decoding for the NE Python (being transliterated rather than translated). Moreover, the MT system transliterates the term Monty that is unknown to the underlying system. Such entities tend to be unseen in the standard news corpora and consequently unknown (UNK) to the MT systems. Using our classifier in such conditions is expected to reduce the domain gap and improve the translation quality.

<table>
<thead>
<tr>
<th>English Input:</th>
<th>The British comedy troupe Monty Python.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline MT:</td>
<td>Alfrqp Alkwmydyp AlbryTAnyp UNK AFEY</td>
</tr>
<tr>
<td>MT+Classifier:</td>
<td>Alfrqp Alkwmydyp AlbryTAnyp mwnty bAywn.</td>
</tr>
</tbody>
</table>

### 4 Related work

A number of efforts have been made to undertake the NE translation problem for different language pairs. Among them some use sequence of phonetic-based probabilistic models to convert names written in Arabic into the English script (Glover-Stalls and Knight, 1998) for transliteration of names and technical terms that occurs in Arabic texts and originate in English. Others rely on spelling-based model that directly maps an English letter sequence into an Arabic one (Al-Onaizan and Knight, 2002a). In a related work, Al-Onaizan and Knight (2002b) describe a combination of a phonetic-based model and a spelling-based one to build a transliteration model to generate Arabic to English name translations. In the same direction, Hassan et al. (2007) extracted NE translation pairs from both comparable and parallel corpora and evaluate their quality in a NE translation system. More recently, Ling et al. (2011) propose a Web-based method that translates Chinese NEs into English. Our work is similar in its general objectives and framework to the work pre-
sented by Hermjakob et al. (2008), which describes an approach for identifying NEs that should be transliterated from Arabic into English during translation. Their method seeks to find a corresponding English word for each Arabic word in a parallel corpus, and tag the Arabic words as either NEs or non-NEs based on a matching algorithm. In contrast, we tackle this problem in the reverse direction (translating/transliterating English NEs into Arabic). We also present a novel binary classifier for identifying NEs that should be translated and those that should be transliterated.

5 Conclusion and future work

We reported our recent progress on building a classifier which decides if an MT system should translate or transliterate a given named entity. The classifier shows a promising performance in both intrinsic and extrinsic evaluations. We believe that our framework can be expanded to new languages if the required data resources and tools (mainly parallel corpus, Named Entity tagger and transliteration engine) are available. We plan to expand the features and apply the classifier to new languages and conduct MT experiments in domains other than news.

6 Acknowledgements

We thank Nizar Habash and colleagues for the MADA, Arabic detokenization and the transliteration similarity software and also their valuable suggestions. We thank anonymous reviewers for their valuable comments and suggestions. This publication was made possible by grants YSREP-1-018-1-004 and NPRP-09-1140-1-177 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References


Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. Recall-