Addressing Noise in Multidialectal Word Embeddings

Alexander Erdmann, Nasser Zalmout, Nizar Habash
Computational Approaches to Modeling Language Lab
New York University Abu Dhabi
United Arab Emirates
{ae1541,nasser.zalmout,nizar.habash}@nyu.edu

Abstract

Word embeddings are crucial to many natural language processing tasks. The quality of embeddings relies on large non-noisy corpora. Arabic dialects lack large corpora and are noisy, being linguistically disparate with no standardized spelling. We make three contributions to address this noise. First, we describe simple but effective adaptations to word embedding tools to maximize the informative content leveraged in each training sentence. Second, we analyze methods for representing disparate dialects in one embedding space, either by mapping individual dialects into a shared space or learning a joint model of all dialects. Finally, we evaluate via dictionary induction, showing that two metrics not typically reported in the task enable us to analyze our contributions’ effects on low and high frequency words. In addition to boosting performance between 2-53%, we specifically improve on noisy, low frequency forms without compromising accuracy on high frequency forms.

1 Introduction

Many natural language processing tasks require word embeddings as inputs, yet quality embeddings require large, non-noisy corpora. Dialectal Arabic (DA), the low register of highly diglossic Arabic (Ferguson, 1959), is problematically noisy. While the high register, Modern Standard Arabic (MSA), is uniform across educated circles in the Arab World, many varieties of DA are not even mutually intelligible (Chiang et al., 2006). The lexical correspondences across four Arab city dialects in Table 1 demonstrate that this variation is not limited to sound change among cognate forms, but involves significant lexical changes due to borrowing, semantic shift, etc.

<table>
<thead>
<tr>
<th>Rabat</th>
<th>Cairo</th>
<th>Beirut</th>
<th>Doha</th>
<th>MSA</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>مطبخة</td>
<td>قوالة</td>
<td>صنورة</td>
<td>طعامه</td>
<td>طعامه</td>
<td>تومو</td>
</tr>
<tr>
<td>طائرة</td>
<td>طائرة</td>
<td>طائرة</td>
<td>طائرة</td>
<td>طائرة</td>
<td>تايم</td>
</tr>
<tr>
<td>لذيء</td>
<td>حلو</td>
<td>طيب</td>
<td>لذيء</td>
<td>لذيء</td>
<td>ذكيم</td>
</tr>
<tr>
<td>لذيء</td>
<td>حلو</td>
<td>طيب</td>
<td>لذيء</td>
<td>لذيء</td>
<td>ذكيم</td>
</tr>
</tbody>
</table>

Table 1: Lexical correspondences between four urban Arabic dialects and MSA.

Seldom written previously, DA is becoming the dominant form of Arabic on social media, yet annotated data are still scarce (Muhammad AbdulMageed and Elaraby, 2018; Israa Alsarsour and Elsayed, 2018; Kareem Darwish and Kallmeyer, 2018). While complex morphology contributes to sparsity in both MSA and DA (Habash, 2010), noise from inter-dialect variation and unstandardized spelling further reduces token-to-type ratios in DA. This limits opportunities to learn accurate vector representations for any given word. Table 2 shows that the MSA token-to-type ratio is over three times larger than DA, controlling for corpus size. This is still not nearly as large as English due to English’s morphological simplicity. Furthermore, the percentage of tokens belonging to low frequency types is three times greater in DA.

Many previous works ignore inter-dialect variation, training dialect agnostic embeddings, yet we show that modeling dialects individually yields
strong performances in a dictionary induction task when noise is systematically addressed. To that end, we make three contributions. First, we describe simple but effective adaptations to word embedding tools to maximize the informative content leveraged in each training sentence. Second, we compare methods for representing disparate dialects in one embedding space, by mapping individual dialects into shared space or learning a joint model of all dialects. Finally, we evaluate our techniques via dictionary induction, showing that two metrics not typically reported are quite informative. In addition to improving accuracy 2-53%, our adaptations specifically boost performance on noisy, low frequency forms without compromising accuracy on high frequency forms.

2 Related Work

Common monolingual embedding models are trained to predict either the target word given the context (Continuous Bag of Words) or elements of the context given the target (SkipGram) (Mikolov et al., 2013a). These have been adapted to incorporate word order (Trask et al., 2015) or subword information (Bojanowski et al., 2016) to model syntax, morphology, etc.

Bilingual embeddings are vector representations of two languages mapped into shared space, such that translated word pairs have similar vectors (Gouws et al., 2015; Luong et al., 2015). They facilitate applications from parallel sentence extraction (Grover and Mitra, 2017) to machine translation (Zou et al., 2013; Cholakov and Kordoni, 2016; Artetxe et al., 2017b) and can be used to improve monolingual embeddings (Faruqui and Dyer, 2014). Bilingual embeddings are learned via one of three methods: mapping both spaces into a shared space (Mikolov et al., 2013b), monolingual adaptation of one language’s embedding space into another’s (Zou et al., 2013), or bilingually training both embeddings simultaneously (AP et al., 2014; Pham et al., 2015). We compare implementations of two state-of-the-art models for mapping embeddings that use the monolingual adaptation technique, as these best suited our data and resources: VecMap (Artetxe et al., 2016, 2017a) and MUSE (Conneau et al., 2017). Both are equipped to learn either via supervision or by iteratively mapping with little or no supervision. Recently, another unsupervised approach leveraging local neighborhood structures was evaluated on French, English, and MSA (Aldarmaki et al., 2017). Such approaches address seed data scarcity, but have not previously been applied to sparse corpora lacking standardized spelling. While we address unstandardized spelling indirectly by learning better embeddings for low frequency types, Zalmout et al. (2018), Abidi and Smaïli (2018), and Dasigi and Diab (2011) attempt to map DA spelling variants to each other.

We are the first to use embeddings for multiple specific DA dialects, though DA embeddings are often used for sentiment analysis (Al Sallab et al., 2015; Altwayan and Tao, 2016). One such work, Dahou et al. (2016), uses pre-built dictionaries to deterministically identify phrases in mixed MSA-DA data before training embeddings. In MSA, embeddings have been used in additional tasks like morphological analysis (Zalmout and Habash, 2017) and POS tagging (Darwish et al., 2017).

3 Data

We adopt Zaidan and Callison-Burch (2011)’s 4-way coarse-grained dialect distinction of Gulf (GLF), Maghrebi (MAG), Egyptian (EGY), and Levantine (LEV). We collect corpora for each dialect by concatenating the relevant dialect identified portion of the following corpora: Almeman and Lee (2013)’s web crawl of forums, comments and blogs, Khalifa et al. (2016)’s Gumar corpus of internet novels, the Broad Operational Language Translation corpus of primarily blogs described in Zbib et al. (2012), the dialectal Arabic travel corpus of Bouamor et al. (2018), Zaidan and Callison-Burch (2011)’s online news commentary corpus, and Jarrar et al. (2014)’s corpus of subtitles and tweets. This results in 1.7 million sentences of EGY, 1.5 million GLF, 1.3 million LEV, and 1.1 million MAG. These corpora are each about 200 times smaller than MSA’s single-domain Gigaword (Parker et al., 2011), with lack of standard-
ized spelling and internal domain inconsistency compounding scarcity with noise.

To map dialects’ embeddings into shared spaces and evaluate dictionary induction, we generate seed and test dictionaries similar to Artetxe et al. (2016). We use MGIZA (Koehn et al., 2007) to align 8,000 sentences from Bouamor et al. (2018)’s travel corpus. It contains 12,000 five-way parallel sentences between the DA varieties of Beirut (LEV), Cairo (EGY), Doha (GLF), Tunis (MAG), and Rabat (MAG), but we collapse Tunis and Rabat to match Zaidan and Callison-Burch (2011)’s granularity and hold out 4,000 sentences for development on downstream tasks. After alignment, we extract unigram translations from 2,000 sentences to form a bidialectal evaluation dictionary. This yields between 2,500 and 4,000 word pairs, with 1.3 to 1.7 average translations per word depending on the dialect pair. Lastly we realign the remaining 6,000 training sentences and extract a seed dictionary. Three annotators jointly evaluated 400 unigram pairs from the LEV–EGY evaluation dictionary. 89% were acceptable translations.

4 Word Embedding Models

We consider the following models for training word embeddings:

**FT** refers to a FASTTEXT (Bojanowski et al., 2016) implementation of SkipGram with 200 dimensions and a context window of 5 tokens on either side of the target word. A word’s vector is the sum of its SkipGram vector and that of all its component character n-grams between length 2 and 6. Since short vowels are not typically written in Arabic, many affixes only consist of a word start/end token and one character. Thus, these character n-gram parameters outperformed the range of 3 to 6 proposed by Bojanowski et al. (2016) for other languages. In preliminary experiments, FT outperformed WORD2VEC models (Mikolov et al., 2013a; Rehůřek and Sojka, 2010) which lack subword information and hence struggle with Arabic’s morphological complexity. We also compared FT to variant implementations with larger and smaller context windows, though FT consistently performed the same or better.

**EXT** refers to an extended FT model where wide and narrow windowed embeddings, sizes 5 and 1 respectively, are trained separately. Resulting vectors are concatenated to build a 400 dimensional model. Given much work demonstrating that narrow context windows capture more syntactic information and wide windows, semantic information (Pennington et al., 2014; Trask et al., 2015; Goldberg, 2016; Tu et al., 2017), component vectors should complement each other, giving the concatenated vector access to a wider range of linguistic information. To ensure that the improvement came from vector concatenation and not simply from having higher dimensional vectors, we built 400 dimension FT models to compare to EXT, but they did not outperform 200 dimensional FT, likely due to sparsity.

**PP+EXT** refers to an EXT model trained on a preprocessed corpus where phrases have been probabilistically identified. To identify phrases, we recurse over each sentence \(C\) times, each time forming bigram phrases from component unigrams (which could have been longer n-grams in previous iterations) depending on the frequencies of the relevant unigrams and bigrams. We implement this step exactly as described in Mikolov et al. (2013c), but then we copy each output sentence \(C\) times and probabilistically decompose the deterministically identified phrases into smaller n-grams. More precisely, for each deterministically identified n-gram phrase, we progress from the first to the \((n−1)^{th}\) gram, randomly splitting the phrase at that point with probability \(\frac{e^n}{\sum_{r=1}^{C} e^r}\).

The final result of the probabilistic phrase identification is \(C\) potentially unique copies of each sentence containing identified n-gram phrases of length \(n \leq 2^R\). We experimented with linear distributions in addition to the exponential one used for phrase splitting, but the exponential performed better. The exponential distribution means that it is less likely to separate at any given potential break point in longer n-grams than in shorter ones.

Like Mikolov et al. (2013c)’s deterministic identification of phrases, PP+EXT avoids training vectors on individual words in non-compositional phrases, yet PP+Ext’s probabilistic nature lets the model learn from multiple perspectives of every word/phrase’s context, with more informative phrase distributions more likely to appear more frequently. Interestingly, identifying phrases can

---

*Using a development set, we found performance to plateau around \(R=5\) and \(C=15\) and thus adopt these parameters, though higher values of \(C\) could in theory marginally boost performance at the expense of runtime.*
be harmful, as our evaluation is performed on unigrams. We implemented a deterministic version of PP+Ext but it did not outperform the baseline FT as too many unigrams were lost in longer phrases. Thus, identifying phrases probabilistically is crucial to PP+Ext’s high performance.

In preliminary experiments, probabilistic phrase identification improved the FT model without extending vectors, yet the performance did not exceed Ext. Hence, we only report PP+Ext scores, as the technique is far more effective when coupled with Ext. The combination of techniques is actually designed to be complementary: FT leverages morphology, Ext combines syntax with semantics, and probabilistic phrase identification increases the number of meaningful contexts used for training. These enable the model to learn better representations for noisy, low frequency forms without requiring additional data.

5 Multidialectal Embedding Space

We consider two options for generating multidialectal embeddings for DA: (a) a dialect agnostic model trained on all DA corpora, and (b) training individual dialect models separately before mapping them into a shared embedding space. While (b) leverages less data per model, (a) is subject to more noise and ambiguity, as many words are unique to certain dialects or have disparate meanings in different dialects. (b) can be seeded with a bidialectal dictionary or parallel sentences. We found the dictionary approach to perform better.

ALLDA is a PP+Ext model trained on a combined corpus of all dialects. To avoid code switching issues, ALLDA assigns words only to those dialects for which its relative frequency in that dialect’s corpus is greater than 5% of its maximum relative frequency in any dialect. Thus, a word assigned to multiple dialects will take the same vector in each dialect and be its own nearest neighbor for any dialect pairs where it belongs to both.

VECMAP is Artetxe et al. (2016, 2017a)’s tool that uses a seed dictionary (or shared numerals) to learn a mapping function which minimizes distances between seed dictionary unigram pairs. In data scarce settings, the function can be learned iteratively, inducing a larger seed dictionary each round, yet the noise in our DA corpora prevents this process from getting off the ground, producing scores of zero after a few iterations.

MUSE is Conneau et al. (2017)’s tool, using adversarial learning (and optionally a seed) to identify similarly behaving high frequency anchor words, bootstrapping into fine tuning the mapping of less frequent words. MUSE is specifically designed for data scarce and unsupervised settings. It assumes shared embedding structures to be identifiable, and the authors demonstrate that domain differences can strain this assumption.

6 Experiments and Results

To evaluate the quality of our DA word embeddings, we use the task of dictionary induction. Given source dialect words from the evaluation dictionary, we attempt to recall appropriate translations in the target dialect based on cosine distance in multidialectal embedding space. The standard metric for this task is precision@k=1 (P@1) (Artetxe et al., 2016, 2017a; Conneau et al., 2017), measuring the fraction of source words in the evaluation dictionary for which the nearest target dialect neighbor matches any of the possible translations in the evaluation dictionary.

We, however, are also concerned with how well multiple translations are recalled, as many words become polysemous in DA with short vowels omitted and spelling not standardized. For this reason, many words appearing both in the seed and evaluation dictionaries do not map to the exact same set of possible translations in each. Thus, many precision errors may be forgivable, so we focus on recall, reporting the metric recall@k=5 (R@5). Lastly, because types appear in a Zipfian distribution and type-based metrics disproportionately reflect accuracy in the tail, we report a frequency weighted recall@k=5 (WR@5) as well.\(^7\) Considering both R@5 and WR@5 avoids the risk of improving performance on high or low frequency types at the expense of the other.

In Table 3, models FT, Ext, and PP+Ext are trained on individual dialects, then mapped using supervised VECMAP into bidialectal embedding spaces. We experimented with all combinations of mapping algorithms and embedding models, yet VECMAP consistently outperformed the other mapping algorithms. We also report results for unsupervised UMUSE leveraging PP+Ext embeddings. ID is an identity dictionary mapping

\(^7\)R@5 and WR@5 are normalized by the score of an oracle that correctly recalls up to 5 translations of every source word, but no more should more exist. Thus, the maximum score for these metrics is 1, making them comparable to P@1.
Table 3: Dictionary induction results comparing various multidialectal embedding models mapped via supervised (SVENMAP) and unsupervised (ALLDA, UMIUSE) techniques.

<table>
<thead>
<tr>
<th>Metric</th>
<th>ID</th>
<th>FT</th>
<th>EXT</th>
<th>PP-EXT</th>
<th>PP+EXT</th>
<th>PP+EXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAG-VECMAP</td>
<td>WR@5</td>
<td>28.9</td>
<td>55.3</td>
<td>42.2</td>
<td>47.0</td>
<td>32.6</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>r@5</td>
<td>24.9</td>
<td>36.2</td>
<td>40.4</td>
<td>51.1</td>
</tr>
<tr>
<td>LEVECMAP</td>
<td></td>
<td>33.6</td>
<td>35.3</td>
<td>39.7</td>
<td>54.0</td>
<td>33.7</td>
</tr>
<tr>
<td>MAG-VECMAP</td>
<td>↓</td>
<td>r@5</td>
<td>37.8</td>
<td>46.9</td>
<td>49.7</td>
<td>50.8</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>p@1</td>
<td>30.4</td>
<td>36.9</td>
<td>41.2</td>
<td>45.2</td>
</tr>
<tr>
<td>GLF-VECMAP</td>
<td></td>
<td>35.0</td>
<td>31.1</td>
<td>37.9</td>
<td>40.0</td>
<td>29.6</td>
</tr>
<tr>
<td>MAG-VECMAP</td>
<td>WR@5</td>
<td>42.2</td>
<td>42.7</td>
<td>48.1</td>
<td>48.3</td>
<td>45.8</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>r@5</td>
<td>30.7</td>
<td>34.5</td>
<td>39.4</td>
<td>42.9</td>
</tr>
<tr>
<td>EGYVECMAP</td>
<td></td>
<td>36.0</td>
<td>29.4</td>
<td>38.0</td>
<td>36.6</td>
<td>36.3</td>
</tr>
<tr>
<td>EGY-VECMAP</td>
<td>WR@5</td>
<td>42.9</td>
<td>51.3</td>
<td>51.3</td>
<td>52.8</td>
<td>47.8</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>r@5</td>
<td>40.9</td>
<td>48.2</td>
<td>49.9</td>
<td>52.8</td>
</tr>
<tr>
<td>GLF-VECMAP</td>
<td></td>
<td>47.7</td>
<td>43.3</td>
<td>48.5</td>
<td>48.3</td>
<td>41.7</td>
</tr>
<tr>
<td>LEVECMAP</td>
<td>WR@5</td>
<td>43.2</td>
<td>50.6</td>
<td>50.4</td>
<td>51.7</td>
<td>48.5</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>r@5</td>
<td>33.6</td>
<td>37.8</td>
<td>38.9</td>
<td>46.4</td>
</tr>
<tr>
<td>GLF-VECMAP</td>
<td></td>
<td>39.0</td>
<td>34.1</td>
<td>37.5</td>
<td>41.7</td>
<td>33.1</td>
</tr>
<tr>
<td>LEVECMAP</td>
<td>WR@5</td>
<td>44.0</td>
<td>50.3</td>
<td>49.8</td>
<td>52.4</td>
<td>50.6</td>
</tr>
<tr>
<td></td>
<td>↓</td>
<td>r@5</td>
<td>33.0</td>
<td>27.6</td>
<td>39.6</td>
<td>42.3</td>
</tr>
<tr>
<td>EGY-VECMAP</td>
<td></td>
<td>39.6</td>
<td>33.8</td>
<td>38.8</td>
<td>37.7</td>
<td>39.2</td>
</tr>
</tbody>
</table>

7 Conclusion and Future Work

We presented techniques for generating multidialectal word embeddings from noisy DA corpora. Due to linguistic differences, modeling dialects independently and mapping embeddings into multidialectal space generally outperformed training dialect agnostic embeddings on combined corpora. Our novel techniques include concatenating narrow and wide windowed vectors and probabilistically identifying phrases before training embeddings. These techniques improved performance on bidialectal dictionary induction 2-53% over a state-of-the-art baseline, with most of the improvement realized on noisy, low frequency word forms. Our approach can easily be applied to other, similarly noisy corpora. In future work, we will improve the handling of orthographically ambiguous words, which are very prevalent in DA, and we will evaluate on the downstream applications of machine translation and morphological disambiguation.

Acknowledgments

The second author was supported by the New York University Abu Dhabi Global PhD Student Fellowship program. We are also grateful for the support of the High Performance Computing Center at New York University Abu Dhabi.
References


Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, A Aziz Altowayan and Lixin Tao. 2016. Word em-


Nasser Zalmout and Nizar Habash. 2017. Don’t throw those morphological analyzers away just yet: Neural
