DiaLex:
A Benchmark for Evaluating Multidialectal Arabic Word Embeddings

Muhammad Abdul-Mageed[1] **Shady Elbassuoni**[2] **Jad Doughman**[2]
AbdelRahim Elmadany[1] **El Moatez Billah Nagoudi**[1] **Yorgo Zoughby**[2]
Ahmad Shaher[1] **Iskander Gaba**[2] **Ahmed Helal**[3] **Mohammed El-Razzaz**[4]
[1]Natural Language Processing Lab, The University of British Columbia, Vancouver, Canada
[2]American University of Beirut, Beirut, Lebanon
[3] Concordia University, Montreal, Canada
[4] Arab Academy for Science and Technology, Cairo, Egypt
[1]{muhammad.mageed,moatez.nagoudi,a.elmadany,ahmad-shaher}@ubc.ca
[2]{sd58,jad17,ytz00,amh90,img02}@aub.edu.lb
[3]amh90@mail.aub.edu [4]mohammed.elrzzaz@gmail.com

Abstract

Word embeddings are a core component of modern natural language processing systems, making the ability to thoroughly evaluate them a vital task. We describe DiaLex, a benchmark for intrinsic evaluation of dialectal Arabic word embeddings. DiaLex covers five important Arabic dialects: Algerian, Egyptian, Lebanese, Syrian, and Tunisian. Across these dialects, DiaLex provides a testbank for six syntactic and semantic relations, namely *male to female*, *singular to dual*, *singular to plural*, *antonym*, *comparative*, and *genitive to past tense*. DiaLex thus consists of a collection of word pairs representing each of the six relations in each of the five dialects. To demonstrate the utility of DiaLex, we use it to evaluate a set of existing and new Arabic word embeddings that we developed. Beyond evaluation of word embeddings, DiaLex supports efforts to integrate dialects into the Arabic language curriculum. It can be easily translated into Modern Standard Arabic and English, which can be useful for evaluating word translation. Our benchmark, evaluation code, and new word embedding models will be publicly available. [1]

1 Introduction

Word embeddings are the backbone of modern natural language processing (NLP) systems. They encode semantic and syntactic relations between words by representing them in a low-dimensional space. Many techniques have been proposed to learn such embeddings (Pennington et al., 2014; Mikolov et al., 2013a; Mnih and Kavukcuoglu, 2013) from large text corpora. As of today, a large number of such embeddings are available in many languages including Arabic. Due to their importance, it is vital to be able to evaluate word embeddings, and various methods have been proposed for evaluating them. These methods can be broadly categorized into *intrinsic* evaluation methods and *extrinsic* evaluation ones. For extrinsic evaluation, word embeddings are assessed based on performance in downstream applications. For intrinsic evaluation, they are assessed based on how well they capture syntactic and semantic relations between words.

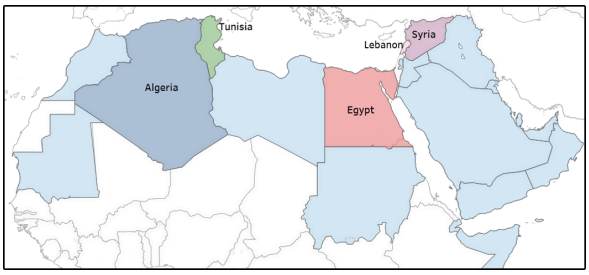

Figure 1: A map of the five Arab countries covered by DiaLex. The five countries cover different regions in the Arab world: two in the western region (Algeria and Tunisia), one in the middle (Egypt), and two in the eastern region (Lebanon and Syria).

Although there exists a benchmark for evaluating modern standard Arabic (MSA) word embeddings (Elrazzaz et al., 2017), no such resource that we know of exists for Arabic dialects. This makes it difficult to measure progress on Arabic dialect processing. In this paper, our goal is to facilitate intrinsic evaluation of dialectal Arabic word embeddings. To this end, we build a new benchmark spanning five different Arabic dialects, from Eastern, Middle, and Western Arab World. Namely, our benchmark covers Algerian (ALG), Egyptian (EGY), Lebanese (LEB), Syr-

[1]https://github.com/UBC-NLP/dialex.

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 11–20
Kyiv, Ukraine (Virtual), April 19, 2021.

ian (SYR), and Tunisian (TUN). Figure 1 shows a map of the five Arab countries covered by DiaLex. For each one of these dialects, DiaLex consists of a set of word pairs that are syntactically or semantically related by one of six different relations: *Male to Female, Singular to Plural, Singular to Dual, Antonym, Comparative*, and *Genitive to Past Tense*. Overall, DiaLex consists of over 3,000 word pairs in those five dialects, evenly distributed. To the best of our knowledge, DiaLex is the first benchmark that can be used to assess the quality of Arabic word embeddings in the five dialects it covers.

To be able to use DiaLex to evaluate Arabic word embeddings, we generate a set of word analogy questions from the word pairs in DiaLex. A word analogy question is generated from two word pairs from a given relation. These questions have recently become the standard in intrinsic evaluation of word embeddings (Mikolov et al., 2013a; Gao et al., 2014; Schnabel et al., 2015). To demonstrate the usefulness of DiaLex in evaluating Arabic word embeddings, we use it to evaluate a set of existing and new Arabic word embeddings. We conclude that both available and newly-developed word embedding models have moderate-to-serious coverage issues and are not sufficiently representative of the respective dialects under study. ***In addition to the benchmark of word pairs, our newly-developed dialectal Arabic word embeddings will also be publicly available***.[2]

Beyond evaluation of word embeddings, we envision DiaLex as a basis for creating multidialectal Arabic resources that can facilitate study of the semantics and syntax of Arabic dialects, for example for pedagogical applications (Mubarak et al., 2020). More broadly, we hope DiaLex will contribute to efforts for integrating dialects in the Arabic language curriculum (Al-Batal, 2017). DiaLex can also be used to complement a growing interest in contextual word embeddings (Peters et al., 2018) and self-supervised language models (Devlin et al., 2019), including in Arabic (Antoun et al., 2020; Abdul-Mageed et al., 2020a; Lan et al., 2020a). Extensions of DiaLex can also be valuable for NLP, for example, DiaLex can be easily translated into MSA, other Arabic dialects, English, or other languages. This extension can enable evaluation of word-level translation systems, including in cross-lingual settings (Aldarmaki et al., 2018; Aldarmaki and Diab, 2019). Our resources can also be used in comparisons against contextual embeddings (Peters et al., 2018) and embeddings acquired from language models such as BERT (Devlin et al., 2019). For example, it can be used in evaluation settings with Arabic language models such as AraBert (Antoun et al., 2020), GigaBERT (Lan et al., 2020b), and the recently developed ARBERT and MARBERT (Abdul-Mageed et al., 2020a). More generally, our efforts are motivated by the fact that the study of Arabic dialects and computational Arabic dialect processing is a nascent area with several existing gaps to fill (Bouamor et al., 2019; Abbes et al., 2020; Abdul-Mageed et al., 2020b, 2021, 2020c).

The rest of the paper is organized as follows. In Section 2, we describe how DiaLex was constructed. Section 3 offers our methodical generation of a testbank for evaluating word embeddings. In Section 4, we provide a case study for evaluating various word embedding models, some of which are newly developed by us. Section 5 is about related work. Section 6 is where we conclude and present future directions.

2 Benchmark Construction

DiaLex consists of a set of word pairs in five different Arabic dialects and for six different semantic and syntactic relations, namely Male to Female, Singular to Plural, Singular to Dual, Antonym, Comparative, and Genitive to Past Tense. We chose only this set of relations as they are standard in previous literature. In addition, they are comprehensive enough to reflect dialect specificity. A good word embeddings model should thus have close representation of word pairs for each of these relations in the embeddings space.

For each dialect, word pairs were *manually* generated by at least one native speaker of the dialect. Each person independently came up with the word pairs representing a given relation based on their knowledge of the dialect and while trying to include words that are typically representative and unique in that dialect. That is, to the best of our ability, the words were chosen so that they are frequently-used words in the dialect and are not the same as in MSA. One challenge we faced when generating the word pairs was orthographic variation. For example, consider the antonym of the

[2]Our benchmark, evaluation code, and new word embedding models will be available at: https://github.com/UBC-NLP/dialex.

Dialect	antonym	compara	genitive-pt	male-fem	sing-pl	sing-dual	all
Algerian	100	100	100	100	102	105	**607**
Egyptian	98	98	998	99	97	98	**588**
Lebanese	98	99	109	96	105	126	**633**
Syrian	98	97	98	101	97	102	**593**
Tunisian	100	100	100	100	102	147	**649**
Total	**494**	**494**	**505**	**496**	**503**	**578**	**3,070**

Table 1: Statistics of DiaLex across different relations. (**Genitive-pt=** genitive-past tense).

word ورا ("behind") in the Egyptian dialect. It can be written as قدام or أودام ("in front of") . We decided to include all variations of the same word in the benchmark. A second challenge was the need to present the relationship using more than one word. For instance, consider Algerian and the relationship dual, the word زوج ("pair") is sometimes used to describe two items of something. So, for instance, for the word ساعه ("hour"), the dual can be either زوج سوايع ("pair of hours") or ساعتين ("two hours"). Again, we opted for including both variants in the benchmark. Overall, for each dialect and each relation, around 100 word pairs were generated. Table 1 shows the statistics of DiaLex's word pairs lists and Table 2 shows some example word pairs in DiaLex and their English and MSA translations. Overall, DiaLex consists of a total of $3,070$ word pairs distributed evenly among the dialects and the relations.

3 Testbank for Evaluating Word Embeddings and Evaluation Metric

Given the word pair lists in DiaLex, we generate a testbank consisting of 260,827 tuples. Each tuple consists of two word pairs (a, b) and (c, d) from the same relation and the same dialect. For each of our five dialects and for each of our six relations, we generate a tuple by combining two different word pairs from the same relation in the same dialect. Once tuples have been generated, they can be used as word analogy questions to evaluate different word embeddings as defined by Mikolov et al. (Mikolov et al., 2013a). A word analogy question for a tuple consisting of two word pairs (a, b) and (c, d) can be formulated as follows: "a to b is like c to ?". Each such question will then be answered by calculating a target vector $t = b - a + c$. We then calculate the cosine similarity between the target vector t and the vector representation of each word w in a given word embeddings V. Finally, we retrieve the most similar word w to t, i.e., $argmax_{w \in V \& w \notin \{a,b,c\}} \frac{w \cdot t}{||w|| ||t||}$. If $w = d$ (i.e., the same word) then we assume that the word embeddings V has answered the question correctly.

Moreover, we extend the traditional word analogy task by taking into consideration if the correct answer is among the top K, with $K \in \{5, 10\}$, closest words in the embedding space to the target vector t, which allows us to more leniently evaluate the embeddings. This is particularly important in the case of Arabic since many forms of the same word exist, usually with additional prefixes or suffixes such as the equivalent of the article "the" or possessive determiners such as "her", "his", or "their". For example, consider one question which asks راجل to ست is like أمير to " ?", i.e., "man" to "woman" is like "prince" to "?", with the answer being "أميره" or "princess". Now, if we rely only on the top-1 word and it happens to be "للأميرة" which means "for the princess" in English, the question would be considered to be answered wrongly. To relax this, and ensure that different forms of the same word will not result in a mismatch, we use the top-5 and top-10 words for evaluation rather than just the top-1.

Note that we consider a question to be answered wrongly if at least one of the words in the question are not present in the word embeddings. That is, we take into consideration the coverage of the embeddings as well (Gao et al., 2014).

Finally, we report the number of questions that were answered correctly over the total numbers of questions available. That is, assume the number of questions for a given dialect and a given relation is n, and assume that a given embeddings model M correctly answered m out of those n questions as explained above. Then, the accuracy of the model

Dialect	male-fem	sing-pl	sing-dual
Algerian	سردوك - دجاجة	كسكروط - زوج كسكروطات	بليغة - زوج بليغات
MSA	ديك - دجاجة	شطيرة - شطيرتان	نعلان - نعل
Eng.	rooster - chicken	sandwich - two sandwiches	shoe - pair of shoes
Egyptian	عسول - عسولة	أوضة - إوض	ودن - ودنين
MSA	لطيف - لطيفة	غرفة - غرف	أذن - أذنين
Eng.	nice (m) - nice (f)	room - rooms	ear - two ears
Lebanese	رجال - مرا	إداحة - اداديح	ملئعة - ملئعتين
MSA	رجل - إمرأة	ولاعة - ولاعات	ملعقة - ملعقتين
Eng.	man - woman	lighter - lighters	spoon - two spoons
Syrian	جردون - فارة	كاتو - كاتويات	زنار - زنارين
MSA	جرذ - فأرة	كعكة - كعكات	حزام - حزامين
Eng.	mouse (m) - mouse (f)	cake - cakes	belt - belts
Tunisian	تحفون - تحفونة	علوش - علالش	ريدو - زوز ريدووات
MSA	وسيم - جميلة	خروف - خرفان	ستار - ستاران
Eng.	handsome - beautiful	sheep (sing) - sheep (pl)	curtain - two curtains

Dialect	antonym	comparative	genitive-past tense
Algerian	قاوي - ضعيف	قرعاج - قرعاج أكثر	حوّس - تحواس
MSA	قوي - ضعيف	فضولي - أكثر فضولا	تنزه - نزهة
Eng.	strong - weak	nosy - nosier	promenade - promenaded
Egyptian	ورا - أودام	وحش - أوحش	عيط - عياط
MSA	خلف - أمام	سئ - أسوأ	بكي - بكاء
Eng.	back - front	bad - worse	cried - crying
Lebanese	مرتاح - مزعوج	جكل - اجكل	فات - فوتة
MSA	مرتاح - منزعج	وسيم - أوسم	دخل - دخول
Eng.	restful - restless	handsome - more handsome	entered - entering
Syrian	دفش - خبط	فهمان - افهم	بحبش - بحبشة
MSA	دفع - اصطدم	ذكي - أذكي	فتش - تفتيش
Eng.	push - bump into	smart - smarter	inspect - inspection
Tunisian	منحوس - مزهار	ميزر - ميزر أكثر	صرفق - تصرفيق
MSA	منكود - محظوظ	فقير - أفقر	صفع - صفع
Eng.	unlucky - lucky	poor - poorer	slap - slapping

Table 2: Example DiaLex word pairs in every dialect across the various relations. For each pair, we also provide MSA and English translations.

M will be $\frac{m}{n}$.

4 Evaluation of Arabic Word Embeddings Using DiaLex

In this section, we demonstrate how DiaLex can be used to evaluate word embeddings across the different dialects it covers. Particularly, we evaluate two large word embeddings models based on Word2Vec (Mikolov et al., 2013b) released by Zahran et al. (2015). One model is based on skip grams (Zah_SG) and the other is a continuous bag-of-words (Zah_CBOW). Both of these models have a vocabulary size of $626,3435$ words. We also create four CBOW Word2Vec models, all of which have 300 word vector embedding dimensions, as we describe next.

4.1 Newly-Developed Word Embedding Models

The following are our four newly-developed dialectal Arabic word embedding models:

Twitter-1B. Our first model was trained using a one billion in-house Arabic tweet collection. All tweets were crawled by putting a bounding box crawler around the whole Arab world. Since this collection is large, we only performed *light* preprocessing on it. This involved removing hashtags, URLs, and reducing consecutive repetitions of the same character into only 2. We then trained a CBOW Word2Vec model using the Python library gensim. We set the minimum word frequency at 100 and a window size of 5 words. This model has a vocabulary size of $929,803$ words.

Twitter-250K-MC50. This model uses the same data as Twitter-1B, yet with stricter preprocessing. Namely, we normalize usually orthographically confused Arabic characters by converting *Alif maksura* to *Ya*, reducing all *hamzated Alif* to plain *Alif*, and removing all non-Arabic characters. We then only keep tweets with length ≥ 5 words. This gives us a total of $223,387,189$ tweets (and hence the name *Twitter-250K*). We then use the same parameters as the 1B model, but we set the minimum count to 50 words (again, hence the name Twitter-250K-MC50 where MC50 meaning minimum count of 50). We acquire a model with a vocabulary size of $536,846$ words.

Twitter-250K-MC100. This model is identical with the Twitter-250K-MC50 model, but uses a minimum count of 100 words when training Word2Vec. This model has a vocabulary size of $202,690$ words.

Twitter-Seeds. We use all unigram entries in our benchmark to crawl Twitter, using the search API. In order to avoid overfitting to our benchmark, we randomly sample only 400 words from it for this process. This gives us about $\sim 3M$ tweets. We then crawl up to $3,200$ tweets from the timelines of all $\sim 300K$ users who have posted the initially collected $3M$ tweets. The resulting collection is at $214,161,138$ tweets after strict cleaning and removal of all tweets of length < 5 words. To train an embedding model on this dataset, we use the same Word2Vec parameters as with Twitter-250K-MC50 (i.e., with a minimum count of 50 words). This model has a vocabulary size of $773,311$ words.

Dialect	Model	K=1	K=5	K=10
ALG	Zah_SG	1.79	6.16	7.96
	Zah_CBOW	3.40	8.51	10.78
	Ours-1B	1.32	3.60	5.53
	Ours-MC-50	3.02	8.25	10.90
	Ours-Seeds	**3.88**	9.19	11.51
	Ours-MC-100	3.87	**11.33**	**15.03**
EGY	Zah_SG	1.90	5.93	8.17
	Zah_CBOW	2.06	6.75	8.98
	Ours-1B	2.09	5.86	8.65
	Ours-MC-50	**4.68**	**11.59**	15.67
	Ours-Seeds	3.21	8.19	11.63
	Ours-MC-100	4.33	11.49	**15.97**
SYR	Zah_SG	1.53	4.14	5.20
	Zah_CBOW	1.86	4.76	6.15
	Ours-1B	1.23	4.036	6.85
	Ours-MC-50	2.49	6.50	8.66
	Ours-Seeds	2.54	5.88	8.02
	Ours-MC-100	**3.14**	**9.39**	**12.97**
LEB	Zah_SG	5.14	10.97	13.69
	Zah_CBOW	6.72	12.88	15.51
	Ours-1B	2.29	5.71	7.70
	Ours-MC-50	3.97	9.71	13.83
	Ours-Seeds	3.96	8.77	11.66
	Ours-MC-100	**5.29**	**12.90**	**17.87**
TUN	Zahran_SG	1.38	6.06	8.26
	Zah_CBOW	2.63	8.19	10.77
	Ours-1B	2.53	5.54	8.47
	Ours-MC-50	3.81	8.07	12.20
	Ours-Seeds	**3.80**	**9.08**	**12.49**
	Ours-MC-100	2.49	7.68	12.14

Table 3: Evaluation of six word embedding models using our benchmark across the five dialects. In most cases, our Twitter-250K-MC100 (Ours-MC-100) achieves the best performance.

4.2 Model Evaluation

We evaluate the two models from Zahran et al. (2015) and our four models described above using DiaLex. Table 3 shows average evaluation re-

Dialect	Relation	K=1	K=5	K=10
ALG	sing-dual	None	None	None
	sing-pl	17.39	38.51	47.20
	gen-pt	0.00	2.52	2.52
	antonym	0.83	5.19	7.91
	comp	None	None	None
	male-fem	10.00	35.00	35.00
	total acc	3.87	11.33	15.03
EGY	sing-dual	13.89	27.78	36.11
	sing-pl	27.62	54.29	60.95
	gen-pt	3.40	8.85	13.55
	antonym	3.02	9.51	13.71
	comp	None	None	None
	male-fem	5.00	15.00	20.00
	total acc	4.33	11.49	15.97
SYR	sing-dual	0.00	0.00	0.00
	sing-pl	7.00	30.00	43.00
	gen-pt	0.46	3.72	4.80
	antonym	0.40	2.81	4.81
	comp	3.83	10.71	14.49
	male-fem	6.89	16.70	23.17
	total acc	3.14	9.39	12.97
LEB	sing-dual	0.00	2.04	6.12
	sing-pl	20.99	41.67	50.31
	gen-pt	1.66	4.92	8.36
	antonym	0.43	5.18	9.31
	comp	13.14	26.72	33.62
	male-fem	5.56	13.89	22.22
	total acc	5.29	12.90	17.87
TUN	sing-dual	0.00	0.00	0.00
	sing-pl	11.44	23.20	30.39
	gen-pt	0.00	4.94	11.11
	antonym	0.41	4.00	7.67
	comp	None	None	None
	male-fem	0.00	0.00	0.00
	total acc	2.48	7.68	12.14

Table 4: Evaluation across all relations for our Twitter-MC-100 model. Values shown as "None" are for relationships where the model did not include any of the word pairs in the question tuples in the model vocabulary. Zeros mean the model includes the words in its vocabulary but no correct answers were returned in the top-K.

sults across the different word relationships. For top-1 performance, one or another of our models scores best. For top-5 results, our Twitter-250K-MC100 (Ours-MC-100) acquires best performance for most dialects. Exceptions are EGY and TUN dialects. Ours-MC-100 also performs best on all but TUN dialect. These results show the necessity of developing dialectal resources for the various varieties and that a model trained on large MSA data such as that of Zahran et al. (2015) is quite sub-optimal.

Table 4 shows per-relation results evaluation of the Twitter-250K-MC100 model. We show all-relations results only for this model for space limitations, and we choose this model since it tends to perform well compared to other models. As Table 4 shows, the model works best on the LEB dialect (17.87 accuracy for top-10) and worst on TUN (12.14 accuracy for top-10). The table also shows that the sing-plural relationship is the one most challenging for the model. Clearly, the dual feature either involves (1) bigrams (in which case these unigram models do not work and hence the "None" values) or (2) different ways of expressing the same meaning of duality. For example, in the LEB pair كوريدور-كوريدورين "corridor-two corridors", duality can be expressed also by the phrase كوريدور ٢, using the digit "2" instead of the dual suffix ين in كوريدورين. Overall, these results suggest that even our developed models are neither sufficiently powerful nor large enough (even with large vocabularies close to 1M words) to cover all the dialects. We also observe that models need to see enough contexts (words $>$ 50 minimum count) to scale well. This calls for the development of more robust models with wider coverage and more frequent contexts. Our Twitter collection of 5M words can be used towards that goal, but we opted for not exploiting it for building a word embeddings model since this would be considered overfitting to our benchmark.

5 Related Work

5.1 Arabic Word Embeddings

The number of available Arabic word embeddings is increasing rapidly. Some of these are strictly trained using textual corpora written in MSA, while others were trained using dialectal data. We review the most popular embedding models we are aware of here.

Zahran et al. (2015) built three models for Arabic word embeddings (CBOW, SKIP-G, and GloVe). To train these models, they used a large collection of MSA texts totaling $\sim$ 5.8B words. The sources used include Arabic Wikipedia, Arabic Gigaword (Parker et al., 2009), Open Source Arabic Corpora (OSAC) (Saad and Ashour, 2010), OPUS (Tiedemann, 2012), MultiUN (Chen and Eisele, 2012), and a few others. Soliman et al. (2017) proposed *AraVec* a set of Arabic word embedding models. It consist of six word embedding models built on top of three different Arabic content domains; Wikipedia Arabic, World Wide Web pages, and Tweets with more than 3.3 billion

word. Both CBOW and SKIP-Gram architecture are investigated in this work.

Abdul-Mageed et al. (2018) build an SKIP-G model using $\sim$ 234M tweets, with vector dimensions at 300. The authors, however, do not exploit their model in downstream tasks. Abu Farha and Magdy (2019) built two word embeddings models (CBOW and SKIP-Gram) exploiting 250M tweets. The authors used the models in the context of training the sentiment analysis system *Mazajak*. The dimensions of each embedding vector in the Mazajak models are at 300.

More recently, El-Haj (2020) developed *Habibi*, a Multi Dialect Multi-National Arabic Song Lyrics Corpus which comprises more than $30,000$ Arabic song lyrics from 18 Arab countries and six Arabic dialects for singers. *Habibi* contains $500,000$ sentences (song verses) with more than 3.5 million words. Moreover, the authors provided a 300 dimension CBOW word embeddings of the corpus. Doughman et al. (2020) built a set of word embeddings learnt from three large Lebanese news archives, which collectively consist of $609,386$ scanned newspaper images and spanning a total of 151 years, ranging from 1933 till 2011. To train the word embeddings, Optical Character Recognition (OCR) was employed to transcribe the scanned news archives, and various archive-level as well as decade-level word embeddings were trained. In addition, models were also built using a mixture of Arabic and English data. For example, Lachraf et al. (2019) presented *AraEngVec* an Arabic-English cross-lingual word embedding models. To train their bilingual models, they used a large dataset with more than 93 million pairs of Arabic-English parallel sentences (with more than 1.8 billion words) mainly extracted from the Open Parallel Corpus Project (OPUS) (Tiedemann, 2012). In order to train the models, they have chosen CBOW and SKIP-Gram as an architecture. Indeed, they propose three methods for pre-processing the opus dataset: parallel sentences, word-level alignment and random shuffling. Both extrinsic and intrinsic evaluations for the different *AraEngVec* model variants. The extrinsic evaluation assesses the performance of models on the Arabic-English Cross-Language Semantic Textual Similarity (CL-STS) task (Nagoudi et al., 2018), while the intrinsic evaluation is based on the Word Translation (WT) task.

Some works have also investigated the utility of using morphological knowledge to enhance word embeddings. For example, Erdmann and Habash (2018) demonstrated that out-of-context rule-based knowledge of morphological structure can complement what word embeddings can learn about morphology from words' in-context behaviors. They quantified the value of leveraging subword information when learning embeddings and the further value of noise reduction techniques targeting the sparsity caused by complex morphology such as in the Arabic language case.

El-Kishky et al. (2019) tackled the problem of root extraction from words in the Semitic language family. They proposed a constrained sequence-to-sequence root extraction method. Furthermore, they demonstrated how one can leverage the root information alongside a simple slot-based morphological decomposition to improve upon word embedding representations as evaluated through word similarity, word analogy, and language modeling tasks. In this paper, we have demonstrated the effectiveness of our benchmark DiaLex in the evaluation of a chosen set of these available word embedding models and compared them to newly-developed ones by us as we explain in the next section.

5.2 Word Embeddings Evaluation

There is a wealth of research on evaluating unsupervised word embeddings, which can be broadly divided into intrinsic and extrinsic evaluations. Intrinsic evaluations mostly rely on word analogy questions and measure the similarity of words in the low-dimensional embedding space (Mikolov et al., 2013a; Gao et al., 2014; Schnabel et al., 2015). Extrinsic evaluations assess the quality of the embeddings as features in models for other tasks, such as semantic role labeling and part-of-speech tagging (Collobert et al., 2011), or noun-phrase chunking and sentiment analysis (Schnabel et al., 2015). However, all of these tasks and benchmarks are built for English and thus cannot be used to assess the quality of Arabic word embeddings, which is the main focus here.

To the best of our knowledge, only a handful of recent studies attempted evaluating Arabic word embeddings. Zahran et al. (2015) translated the English benchmark in (Mikolov et al., 2013a) and used it to evaluate different embedding techniques when applied on a large Arabic corpus. However,

as the authors themselves point out, translating an English benchmark is not the best strategy to evaluate Arabic embeddings. Zahran et al. (2015) also consider extrinsic evaluation on two NLP tasks, namely query expansion for Information Retrieval and short answer grading.

Dahou et al. (2016) used the analogy questions from Zahran et al. (2015) after correcting some Arabic spelling mistakes resulting from the translation and after adding new analogy questions to make up for the inadequacy of the English questions for the Arabic language. They also performed an extrinsic evaluation using sentiment analysis. Finally, Al-Rfou et al. (2013) generated word embeddings for 100 different languages, including Arabic, and evaluated the embeddings using part-of-speech tagging, however the evaluation was done only for a handful of European languages. Elrazzaz et al. (2017) built a benchmark in MSA that can be utilized to perform intrinsic evaluation of different word embeddings using word analogy questions. They then used the constructed benchmark to evaluate various Arabic word embeddings. They also performed extrinsic evaluation of these word embeddings using two NLP tasks, namely Document Classification and Named Entity Recognition.

Salama et al. (2018) investigated enhancing Arabic word embeddings by incorporating morphological annotations to the embeddings model. They tuned the generated word vectors to their lemma forms using linear compositionality to generate lemma-based embeddings. To assess the effectiveness of their model, they used the benchmark built by Elrazzaz et al. (2017). Taylor and Brychcín (2018b) demonstrated several ways to use morphological Arabic word analogies to examine the representation of complex words in semantic vector spaces. They presented a set of morphological relations, each of which can be used to generate many word analogies.

El Bazi and Laachfoubi (2017) investigated the effect of stemming on Arabic word representations. They applied various stemmers on different word representations approaches, and conducted an extrinsic evaluation to assess the quality of these word vectors by evaluating their impact on the Named Entity Recognition task for Arabic. Taylor and Brychcín (2018a) provided a corpus of Arabic analogies focused on the morphological constructs which can participate in verb, noun and prepositional phrases. They conducted an examination of ten different semantic spaces to see which of them is most appropriate for this set of analogies, and they illustrated the use of the corpus to examine phrase-building. Barhoumi et al. (2020) proposed intrinsic and extrinsic methods to evaluate word embeddings for the specific task of Arabic sentiment analysis. For intrinsic evaluation, they proposed a new method that assesses what they define as the "sentiment stability" in the embedding space. For extrinsic evaluation, they relied on the performance of the word embeddings to be evaluated for the task of sentiment analysis. They also trained various word embeddings using different types of corpora (polar and non-polar) and evaluated them using their proposed methods. To the best of our knowledge, our proposed benchmark is the first benchmark developed in various Arabic dialects and that can be used to perform intrinsic evaluation of Arabic word embedding with respect to those dialects.

6 Conclusion

We described DiaLex, a benchmark for evaluating dialectal Arabic word embeddings. DiaLex comes in five major Arabic dialects, namely Algerian, Egyptian, Lebanese, Syrian, and Tunisian. Across these dialects, DiaLex offers a testbank of word pairs for six syntactic and semantic relations, namely *male to female*, *singular to dual*, *singular to plural*, *antonym*, *comparative*, and *genitive to past tense*. To demonstrate the utility of DiaLex, we used it to evaluate a set of available and newly-developed Arabic word embedding models. Our evaluations are intended to showcase the utility of our new benchmark. DiaLex as well as the newly-developed word embeddings will be publicly available.

DiaLex can be used to support integration of dialects in the Arabic language curriculum, and for the study of the syntax and semantics of Arabic dialects. It can also complement evaluations of contextual word embeddings. In the future, we plan to use DiaLex to for more extensive evaluation of all publicly-available Arabic word embedding models. We will also train a dialectal Arabic word embeddings model with a larger dataset and evaluate it using DiaLex. Finally, we will translate DiaLex into MSA and English to facilitate use of the resource for evaluating word translation.

References

Ines Abbes, Wajdi Zaghouani, Omaima El-Hardlo, and Faten Ashour. 2020. Daict: A dialectal Arabic irony corpus extracted from twitter. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6265–6271.

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *LREC*, pages 3653–3659.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020a. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. *arXiv preprint arXiv:2101.01785*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020b. Nadi 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The Second Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP 2021)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, and Lyle Ungar. 2020c. Micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

Mahmoud Al-Batal. 2017. *Arabic as one language: Integrating dialect in the Arabic language curriculum*. Georgetown University Press.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Hanan Aldarmaki and Mona Diab. 2019. Context-aware cross-lingual mapping. *arXiv preprint arXiv:1903.03243*.

Hanan Aldarmaki, Mahesh Mohan, and Mona Diab. 2018. Unsupervised word mapping using structural similarities in monolingual embeddings. *Transactions of the Association for Computational Linguistics*, 6:185–196.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for Arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Amira Barhoumi, Nathalie Camelin, Chafik Aloulou, Yannick Estève, and Lamia Hadrich Belguith. 2020. Toward qualitative evaluation of embeddings for arabic sentiment analysis. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4955–4963.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19), Florence, Italy*.

Yu Chen and Andreas Eisele. 2012. Multiun v2: Un documents with multilingual alignments. In *LREC*, pages 2500–2504.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Abdelghani Dahou, Shengwu Xiong, Junwei Zhou, Mohamed Houcine Haddoud, and Pengfei Duan. 2016. Word embeddings and convolutional neural network for arabic sentiment classification. In *International Conference on Computational Linguistics*, pages 2418–2427.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jad Doughman, Fatima Abu Salem, and Shady El-bassuoni. 2020. Time-aware word embeddings for three lebanese news archives. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4717–4725.

Ismail El Bazi and Nabil Laachfoubi. 2017. Is stemming beneficial for learning better arabic word representations? In *First International Conference on Real Time Intelligent Systems*, pages 508–517. Springer.

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

Ahmed El-Kishky, Xingyu Fu, Aseel Addawood, Nahil Sobh, Clare Voss, and Jiawei Han. 2019. Constrained sequence-to-sequence semitic root extraction for enriching word embeddings. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 88–96.

Mohammed Elrazzaz, Shady Elbassuoni, Khaled Shaban, and Chadi Helwe. 2017. Methodical evaluation of arabic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–458.

Alexander Erdmann and Nizar Habash. 2018. Complementary strategies for low resourced morphological modeling. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 54–65.

Bin Gao, Jiang Bian, and Tie-Yan Liu. 2014. Wordrep: A benchmark for research on learning word representations. *arXiv preprint arXiv:1407.1640*.

Raki Lachraf, El Moatez Billah Nagoudi, Youcef Ayachi, Ahmed Abdelali, and Didier Schwab. 2019. ArbEngVec : Arabic-English cross-lingual word embedding model. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 40–48, Florence, Italy. Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020a. An empirical study of pre-trained transformers for arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020b. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

Hamdy Mubarak, Shimaa Amer, Ahmed Abdelali, and Kareem Darwish. 2020. Arabic curriculum analysis. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 80–86.

El Moatez Billah Nagoudi, Jérémy Ferrero, Didier Schwab, and Hadda Cherroun. 2018. Word embedding-based approaches for measuring semantic similarity of arabic-english sentences. In *Arabic Language Processing: From Theory to Practice*, pages 19–33, Cham. Springer International Publishing.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2009. Arabic gigaword.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Motaz K Saad and Wesam Ashour. 2010. Osac: Open source arabic corpora. In *6th ArchEng Int. Symposiums, EEECS*, volume 10.

Rana Aref Salama, Abdou Youssef, and Aly Fahmy. 2018. Morphological word embedding for arabic. *Procedia computer science*, 142:83–93.

Tobias Schnabel, Igor Labutov, David M Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *EMNLP*, pages 298–307.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Stephen Taylor and Tomáš Brychcín. 2018a. Arabic word analogies and semantics of simple phrases. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–6. IEEE.

Stephen Taylor and Tomáš Brychcín. 2018b. The representation of some phrases in arabic word semantic vector spaces. *Open Computer Science*, 8(1):182–193.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer.