Persian Word Embedding Evaluation Benchmarks

Mohammad Sadegh Zahedi
Research Fellow
Information Technology Department
ICT Research Institute
Tehran, I.R. Iran
s.zahedi@itrc.ac.ir

Mohammad Hadi Bokaei
Faculty Member
Information Technology Department
ICT Research Institute
Tehran, I.R. Iran
mh.bokaei@itrc.ac.ir

Farzaneh Shoeleh
Research Fellow
Information Technology Department
ICT Research Institute
Tehran, I.R. Iran
f.shoeleh@itrc.ac.ir

Mohammad Mehdi Yadollahi
Research Fellow
Information Technology Department
ICT Research Institute
Tehran, I.R. Iran
mm.yadollahi@itrc.ac.ir

Ehsan Doostmohammadi
Graduate Student
Computational Linguistics Group
Sharif University of Technology
Tehran, I.R. Iran
e.doostm72@student.sharif.edu

Mojgan Farhoodi
Faculty Member
Information Technology Department
ICT Research Institute
Tehran, I.R. Iran
farhoodi@itrc.ac.ir

***Abstract*—Recently, there has been renewed interest in semantic word representation also called word embedding, in a wide variety of natural language processing tasks requiring sophisticated semantic and syntactic information. The quality of word embedding methods is usually evaluated based on English language benchmarks. Nevertheless, only a few studies analyze word embedding for low resource languages such as Persian. In this paper, we perform such an extensive word embedding evaluation in Persian language based on a set of lexical semantics tasks named analogy, concept categorization, and word semantic relatedness. For these evaluation tasks, we provide three benchmark data sets to show the strengths and weakness of five well-known embedding models which are trained on Wikipedia corpus. The experimental results indicates that FastText(sg) and Word2Vec(cbow) outperform other models.**

***Index Terms*—Word Embedding, Evaluation Benchmark, Word2Vec, GloVe, FastText**

I. Introduction

Being inspired by distributional hypothesis of Harris [1] several methods have been developed for automatic extraction of contextual features, the majority of which can be described as a word-context matrix. Primary methods suffered from treating words as atomic individuals, meaning that they could not capture semantic similarities and dissimilarities between words. As most Natural Language Processing (NLP) tasks involve working with words, the field could benefit from methods with more generalized representations. Methods such as LSA and LDA compensated for the shortcoming, until with the work of Bengio et al. [2] "word embedding" came into existence. Later, the idea helped Mikolov et al. [3] to get the task further, this time by using neural networks, which made word embedding a strong trend in NLP.

Word embedding is an efficient method for learning high dimensional distributed word vectors which are able to capture syntactic and semantic relationships. As pointed out by Bengio et al. [2] the main advantage of this model is the use of distributional representation that achieves a level of generalization which helps to overcome the shortcomings of traditional one-hot counterpart. In the one-hot representation, words are discrete units which have no useful relation to each other. On contrary, in word embedding models words which share similar semantic or syntactic features, tend to have similar vectors, which is extensively useful in NLP tasks, as most of them work with words.

In Persian, as in other languages, we need high quality word vectors to be used in various NLP tasks. Such vectors enable us to calculate the similarity between synonymous (and non-synonymous) words such as دکتر and پزشک, the similarity of which is 0 using one-hot vectors. These kind of vectors are most commonly used to compute semantic or syntactic similarities between words, sentences or documents, which is useful in numerous tasks like text summarization [4], machine translation [5] and opinion mining [6].

In this work we have focused on creating standard evaluation benchmarks with their associated test sets to help the community to compare proposed methods in a systematic and standard way. While many such test sets are available for English and many other languages, Persian community suffers from lacking those standard benchmarks. We believe that this work will help the researchers to work more on this field, which is a very important pre-processing step in most tasks. We have also used the latest state of the art algorithms to train Persian word representations and compared them according to these benchmarks. All codes and benchmarks are freely available[1].

[1] http://parsigan.ir

The paper is organized as follows: In the next section, the previous work on word representation and also standard evaluation benchmarks are reviewed. In Section 3, our proposed benchmarks and their associated test sets are introduced. Section 4 specifies our experimental setup including the way we create and normalize training set. Section 5 summarized the results on various tasks and finally the paper is concluded in Section 6.

II. Previous Work

A. Word embedding approaches

Recently, researchers have shown an increased interest in distributed word representations also called word embedding, in the area of natural language processing (NLP). Word embedding exploits statistical properties of textual structure to embed words in a continuous vector space. In this space, terms with similar meanings tend to be located close to each other [7].

Word embedding learning has been used in a variety of NLP tasks such as named entity recognition [8], dependency parsing [9], sentiment analysis [10], information extraction [11], question retrieval [12], question answering [13], [14], [15], etc.). The basic idea is that similar words tend to be close to each other with the vector representation.

There are two basic approaches currently being adopted in research into distributed word representations. One is the matrix factorization approach (also called counter-based methods) and the other is shallow window-based methods (also called prediction-based methods). The matrix factorization methods utilize low-rank approximations to decompose large matrices that capture statistical information about a corpus such as the Latent semantic analysis (LSA) [16]–[18], the Hyperspace Analogue to Language (HAL) [19]. Another approach is to learn word representations that aids in making predictions within local context windows such as word2vec [20], Glove [21] and FastText [22].

LSA, is one of the most used methods for word meaning representation. LSA takes as input a training corpus, i.e. a collection of documents. A word by document co-occurrence matrix is constructed. Typically, normalization is applied to reduce the weight of uninformative high-frequency words in the words-documents matrix. Finally, a dimensionality reduction is implemented by a truncated Singular Value Decomposition, SVD, which projects every word in a subspace of a predefined number of dimensions. In contrast, the HAL utilizes word by word co-occurrence matrix, i.e., the rows and columns correspond to words and the entries correspond to the number of times a given word occurs in the context of another given word.

Recently, a series of works applied deep learning techniques to learn high-quality word representations. Bengio et al. [2] proposed a probabilistic neural network language model (NNLM) for word representations. Furthermore, Mikolov et al. [20] proposed efficient neural network models for learning word representations, including the skip-gram model and the continuous bag-of-word model (cbow), both of which are unsupervised models learned from large-scale text corpora. In both models, a window of predefined length is moved along the corpus, and in each step the network is trained with the words inside the window. Whereas the cbow model is trained to predict the word in the center of the window based on the surrounding words, the skip-gram counterpart is trained to predict the contexts based on the central word. Once the neural network has been trained, the learned linear transformation in the hidden layer is taken as the word representation.

Methods like cbow and skip-gram poorly utilize the statistics of the corpus since they train on local context within the window instead of on global co-occurrence count. In [21], a method named GloVe, for Global Vector, has been proposed. The authors argued that their global log-bilinear regression method are appreciate to produce linear directions of meaning. GloVe as a log-bilinear model with a weighted least-squares objective is trained on the global word-word co-occurrence counts and thus makes efficient use of statistics. The main intuition underlying this model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well.

In [22], Facebook research team proposed a method and released a library named FastText for both learning word representation and sentence classification. FastText differs in the sense that other word representation methods such as skip-gram, cbow and GloVe treat every single word as the smallest unit whose vector representation is to be found. However, FastText assumes a word to be formed by an n-grams of characters, for example, sunny is composed of [sun, sunn,sunny],[sunny,unny,nny] etc, where n could range from 1 to the length of the word.

It is worth mentioning that the only work done on Persian is the work of Bojanowski et al. [22] which is done on 294 languages whose models are available on github[2]. We tested their Persian model which suffered from unnormalized inputs, e.g. البرز and «البرز» are considered two separate words, which resulted in poor quality of the model.

[2]https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

B. Evaluation Methodologies

Word embedding evaluation methods could be divided to intrinsic methods and extrinsic ones. One intrinsic method of evaluation is that of Mikolov et al. [3]. In this method, a list of *analogy questions* is prepared to evaluate the training set. Every line in the test set contains four words, the forth of which is unknown to the testing algorithm at first. After normalizing the first three vectors, w is calculated as $w = v_2 - v_1 + v_3$, the search for the vector whose cosine similarity to w is the greatest begins. This method is similar to the work of Turney [23] that uses multiple-choice analogy questions to evaluate the dual-space model of relational similarity, $sim(a : b, c : d)$.

Some other interesting intrinsic methods include *Semantic Relatedness* which is assessing the correlation between the average human assigned rates on similarity of a pair and the cosine similarity between them; *Synonym Detection* which tries to find synonym of a given word in multiple-choice TOEFL questions using cosine similarity; *Concept Categorization* which reproduces gold categories by clustering vectors and then measures the extent to which each cluster contains concepts from a single gold category; and *Selectional Preferences* which determines how typical a noun is for a verb either as a subject or as an object [24].

As of extrinsic evaluations, the effectiveness of the representation is measured by evaluating how much improvement is achieved in a specific downstream task, when we use the new representation of the words. Of these kind of methods we could mention Syntactic Chunking, Sentiment Classification [25], Part-Of-Speech Tagging, and Named Entity Recognition [26] among other NLP tasks.

III. Persian Word Embedding Evaluation Benchmarks

In order to construct a solid benchmark set to evaluate existing word embedding algorithms on Persian language, we selected three most popular intrinsic tasks, namely analogy, categorization, and semantic relatedness. For each task, we created an appropriate test set, which are introduced in the following.

Analogy The test set of analogy questions comprises of two subsets; syntactic and semantic. These subsets describe word w_1 to be related to word w_2 in the way that word w_3 is related to word w_4. The semantic subset contains family relations, countries and their currencies, Iran's provinces and their capital cities, and countries and their capital cities. An example for each entry in this subset is illustrated in Table I.

The syntactic subset contains adjectives and the adverbs derived from them, nouns and the adverbs derived from them, nouns and their opposites (using prefixes), adjectives and their comparative forms, adjectives and their superlative forms, nationalities, nouns and their plural forms (regular and irregular), the infinitive and third person singular present continuous forms of verbs, the infinitive and third person singular past forms of verbs, third person singular past and third person plural past, and first person singular and first person plural past form of verbs. Most instances were chosen based on frequentness, but many exceptions were included as well. An example for each entry in this subset is illustrated in Table II.

TABLE I
Semantic Subset Examples

Rel. Type	Word Pair 1		Word Pair 2	
Family Rel.	بابا /bɑbɑ/ father	مامان /mɑmɑn/ mother	عمو /ʔamu/ uncle	عمه /ʔamme/ aunt
Currency	روسیه /rusije/ Russia	روبل /rubl/ Ruble	ایران /irɑn/ Iran	ریال /rial/ Rial
Prv. Capit.	فارس /fɑrs/ Fars	شیراز /ʃirɑz/ Shiraz	البرز /alborz/ Alborz	کرج /karɑʤ/ Karaj
Cnt. Capit.	چین /ʧin/ China	پکن /pekan/ Beijing	ایران /irɑn/ Iran	تهران /tehrɑn/ Tehran

There are 30k questions in our test set in different categories. The total number of questions in each category is shown in Table III.

Concept Categorization One of the basic goal of word representation is that similar words tend to be close to each other according to the vector representation. To show how a word representation model is successful to reach this goal, the categorization task is defined. Given a set of nominal concepts, the categorization task is to group them into natural categories, for example football and volleyball should go to the game category, cat and dog into the animal category. To the best of our knowledge, there is no standard categorization benchmark in the Persian language. In this paper, we provide a categorization data set containing 329 Persian concepts organized into 20 categories. Table IV shows the categories and also number of instances in each category.

Word Semantic Relatedness For this task, we use *SemEval2017-Task2* data set[3] which is proposed for Persian language [27]. This data set as a high-quality and well-balanced dataset composed of 500 nominal pairs that are manually scored according to a well-defined similarity scale which asking human subjects to rate the degree of semantic similarity or relatedness between two words on a numerical scale. This dataset includes multi-word expressions, domain-specific terms, and named entities. Since word embedding models can not represent multi-word expressions, we eliminate this type of records from *SemEval2017-Task2*. Hence, our word semantic relatedness data set consists 347 word pairs with their relatedness scores.

[3] http://alt.qcri.org/semeval2017/task2/

TABLE II
Syntactic Subset Example

Rel. Type	Word Pair 1		Word Pair 2	
Adj.-Adv.	سریع /sariʔ/ fast	سریعا /sariʔan/ fast(adv.)	جدا /ʤodɑ/ separate	جداگانه /~gɑne/ separately
N.-Adv.	اصل /asl/ origin	اصلا /aslan/ originally	شب /ʃab/ night	شبانه /ʃabɑne/ nightly
Antonym	توانا /tavɑnɑ/ able	ناتوان /nɑtavɑn/ unable	امن /ʔamn/ safe	ناامن /nɑʔamn/ unsafe
Comp.	سریع /sariʔ/ fast	سریعتر /~tar/ faster	به /beh/ good	بهتر /behtar/ better
Super.	سریع /sariʔ/ fast	سریعترین /~tarin/ fastest	به /beh/ good	بهترین /-tarin/ best
Nat.	ایران /irɑn/ Iran	ایرانی /irɑni/ Iranian	مصر /mesr/ Egypt	مصری /mesri/ Egyptian
Sing.-Pl.	درخت /deraxt/ tree	درختان /~ɑn/ trees	قاضی /gɑzi/ judge	قضات /gozɑt/ judges
1st Pers.	رفتم /raftam/ (I)went	رفتیم /raftim/ (we)~	کردم /kardam/ (I)did	کردیم /kardim/ (we)~
3rd Pers.	رفت /raft/ he* went	رفتند /raftand/ (they)~	کرد /kard/ he did	کردند /kardand/ (they)~
Inf.-Past	رفتن /raftan/ to go	رفت /raft/ (I)went	کردن /kardan/ to do	کرد /kard/ (I)did
Inf.-Pres.	رفتن /raftan/ to go	می‌رود /miravad/ he goes	کردن /kardan/ to do	می‌کند /mikonad/ he does

* There is no gender separation in Persian.

TABLE III
Number of Analogy Questions

Relationship Type	Number	Relationship Type	Number
Family Relationship	342	Currency	1260
Country-Capital	5402	Province-Capital	7832
Adjective-Adverb	1332	Noun-Adverb	1056
Antonym	1260	Comparative	1260
Superlative	1260	Nationality	1406
Singular-Plural	2550	1st Person	1260
3rd Person	1332	Infinitive-Past	1260
Infinitive-Present	1260		
Summation	30072		

IV. Experimental Setup

In order to train word embedding vectors, we need a large corpus of sentences. In this work we used the latest dump of the Persian articles of Wikipedia [4]. We applied several preprocessing steps on this corpus. Characters are normalized and standardized to the fix set of 42 characters (32 Persian alphabet + 10 digits). All other characters are either removed or converted to one of our standard characters. The whole text is also tokenized into different sentences and words. Words with occurrence count less than 5 is also disregarded from the ongoing processes. The whole dataset contains 56,401,975 tokens, 674,465 token types, and 2,943,207 sentences. However, after removing rare words, there are 175k token types which construct the vocabulary.

We examine various state of the art approaches to train the embedding vectors. Specifically we use both skip-gram (sg) and continuous-bag-of-words (cbow) in the word2vec package[5]. We also train GloVe vectors[6] and also The most recent approach which is FastText (according to both cbow and sg algorithms)[7]. For all these approaches, we evaluated various dimensions (50, 100, 300) and various window lengths (2, 4, 6, 8, 10). for other parameters, we used the default ones included in the packages.

After training vectors, the benchmarks introduced in the Section III are used to compare the methods with each other and also with the only available pre-trained Persian vectors trained on the same Wikipedia corpus (without any pre-processing and normalization steps) according to the FastText algorithm with default parameters [22].

TABLE IV
Categorization Data Set Details

Category Name	Count	Category Name	Count
Animal	16	Assets	13
Atmospheric phenomenon	15	Chemical element	20
Creator	13	District	15
Edible fruit	15	Feeling	15
Game	15	Illness	21
Legal document	13	Monetary unit	20
Pain	14	Physical property	20
Social occasion	17	Social unit	20
Solid	19	Time	16
Tree	14	Vehicle	18
Summation	329		

V. Results and Discusion

A. Analogy Task

In order to evaluate the algorithms on the analogy task, for each question $\{w_1, w_2, w_3, w_4\}$ in the set, we calculate the vector $w_2 - w_1 + w_3$. We then find the neighbor set to be the top k nearest words to the calculated vector (we consider k to be 5). If the word w_4 is included in the neighbor set we mark that question as correct, otherwise as wrong. We then calculate accuracy as the ratio of the correct word sets to the number of all word sets in the dataset. The results are summarized in Table V. From this table, it can be seen that the best result (45.37) is achieved with FastText(sg) when the window length is 10 and the new space dimension is 300. The previous pre-trained vector [22] on the same dataset has the accuracy

[4]https://dumps.wikimedia.org/fawiki/

[5]https://radimrehurek.com/gensim/

[6]http://nlp.stanford.edu/projects/glove/

[7]https://fasttext.cc/

of 35.08. This improvement is due to the role of the pre-processing step of the train set.

TABLE V
RESULTS OF THE EVALUATED ALGORITHMS ACCORDING TO THE ANALOGY TASK.

Algorithm	W	D = 50	D = 100	D = 300
Word2Vec(sg)	2	25.49	31.35	32.91
	4	27.64	34.35	36.89
	6	27.30	35.36	38.58
	8	27.74	**35.74**	39.16
	10	**27.82**	35.71	**39.48**
Word2Vec(cbow)	2	22.38	28.15	31.21
	4	25.82	32.95	37.04
	6	25.57	35.58	40.93
	8	28.57	36.86	42.00
	10	**29.57**	**37.40**	**43.96**
GloVe	2	12.63	18.48	21.23
	4	18.64	27.81	32.05
	6	21.24	30.70	35.78
	8	22.53	32.89	37.69
	10	**22.55**	**33.75**	**39.25**
FastText(sg)	2	30.93	39.07	36.23
	4	31.39	40.81	41.35
	6	**32.39**	**41.15**	42.02
	8	32.05	41.10	43.56
	10	29.50	40.05	**45.37**
FastText(cbow)	2	29.63	35.17	31.93
	4	30.14	36.39	33.60
	6	30.84	36.74	34.10
	8	**31.06**	36.76	**34.39**
	10	30.99	**37.21**	34.32

B. Concept Categorization Task

To evaluate the word embedding methods, we firstly run K-means algorithm on the obtained vectors of words in the categorization data set and then calculate the purity measure. To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned vectors and dividing by total number of vectors. Table VI summarizes the results of this task over five models. As results shows, Word2Vec(cbow) with window length of 2 and the space dimension of 100 achieved 74.41 as the best result. This model performs better than the previous pretrained vector presented in [22] on the same dataset which has the purity of 69.70.

C. Words Semantic Relatedness Task

Table VII shows results on Persian word relatedness dataset. A similarity score is obtained from the word vectors by first normalizing each feature across the vocabulary and then calculating the cosine similarity. We compute Spearman's rank correlation coefficient between this score and the human judgments. The result of FastText(sg) with window length of 4 and the vector dimension of 300 is the best one and equals to 68.89. like other tasks, this results is also better than the previous pertained vector [22] on the same dataset which has the correlation score of 66.58.

TABLE VI
RESULTS OF THE EVALUATED ALGORITHMS ACCORDING TO THE CATEGORIZATION TASK.

W	wod2vec (sg)	wod2vec (cbow)	Glove	FastText (sg)	FastText (cbow)	D
2	69.71	70.29	58.82	64.71	**64.71**	50
4	**67.35**	65.88	60.29	62.65	65	
6	65.29	70.59	60.59	63.82	61.47	
8	63.82	**71.18**	**61.76**	65.29	64.41	
10	60	67.65	60.59	**66.18**	64.41	
2	66.47	**74.41**	55	62.35	**64.12**	100
4	**67.94**	68.53	60.29	64.41	62.65	
6	63.24	70	62.94	**64.71**	62.94	
8	62.06	63.24	62.06	60.29	62.06	
10	60.88	65.88	**67.65**	61.47	58.53	
2	62.06	64.41	53.82	**67.35**	**62.65**	300
4	60	**69.71**	55.59	57.06	62.35	
6	56.76	66.18	**60**	62.35	61.76	
8	62.35	67.65	47.94	59.71	60.88	
10	**63.82**	66.76	55.29	63.24	60	

TABLE VII
RESULTS OF THE EVALUATED ALGORITHMS ACCORDING TO THE RELATEDNESS TASK.

W	wod2vec (sg)	wod2vec (cbow)	Glove	FastText (sg)	FastText (cbow)	D
2	**64.05**	61.29	44.79	**65.68**	**59.75**	50
4	63.75	**62.19**	49.29	63.97	59.19	
6	63.31	61.55	51.29	64.59	57.41	
8	62.48	62.16	50.93	63.89	57.62	
10	62.96	62.01	**51.43**	64.09	57.64	
2	64.73	63.81	47.6	**67.83**	**62.07**	100
4	**65.52**	63.34	51.76	66.04	59.77	
6	65.12	63.51	53.83	65.99	59.40	
8	64.77	**64.14**	53.95	65.31	58.7	
10	64.82	62.9	**54.57**	66.41	59.24	
2	63.19	**64.99**	50.55	66.63	**62.23**	300
4	63.85	64.52	53.75	**68.89**	60.5	
6	64.75	63.11	53.96	68.16	59.16	
8	**66.24**	64.25	**56.19**	68.12	59.63	
10	65.5	63.45	55.51	67.9	58.54	

VI. CONCLUSION AND FUTURE WORK

The past decade has seen the rapid development and usage of word embedding in many NLP tasks. Despite the growing interest in vector representations of semantic information, there has been relatively little work on direct evaluations of these models in low resource languages. To the best of our knowledge, there is no study to evaluate word embedding models in Persian language. This paper has presented the first systematic comparative evaluation of word embedding models in Persian language. To do so, we provide three categories of evaluation data set, each of which each focuses on a specific task, named analogy, concept categorization and word semantic relatedness. Our evaluation is done on five state of the art word embedding models, i.e Word2Vec(sg), Word2Vec(cbow), GloVe, FastText(sg), FastText(cbow). The only available pre-trained model on Persian is FastText(sg) model which is trained on Wikipedia corpus in 294 different languages including Persian. However, the limitation of their Persian

model is on un-normalized inputs which results in poor quality. Hence, we first train the word embedding models on normalized Wikipedia corpus with different values of hyper parameters, then evaluate these models over the aforementioned benchmarks. The current study found that the word representation's quality of trained models depends on both the evaluation task and the value of their hyper parameters. The experimental results shows that the FastText(sg) model with $w = 10$ and $d = 300$ has the best performance in analogy task. Word2Vec(cbow) model with $w = 2$ and $d = 100$ and FastText(sg) model with $w = 4$ and $d = 300$ have the best performance in categorization and word semantic relatedness tasks, respectively.

This study has two limitations using only intrinsic evaluation methods and one small size of training corpus. We plan to remedy them and extend the evaluation in future by considering extrinsic evaluation and enriching training corpus by crawling Persian webpages.

Acknowledgment

This study is conducted by support of Iran Telecommunication Research Center. We appreciate *Webazma* team members who have helped us most throughout our research.

References

[1] Z. Harris, "Distributional structure. word 10: 146-162. reprinted in j. fodor and j. katz," *The structure of language: Readings in the philosophy of language*, pp. 775–794, 1954.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[3] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *hlt-Naacl*, vol. 13, 2013, pp. 746–751.

[4] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, 2014, pp. 31–39.

[5] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[6] P. Liu, S. R. Joty, and H. M. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings." in *EMNLP*, 2015, pp. 1433–1443.

[7] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[8] A. Das, D. Ganguly, and U. Garain, "Named entity recognition with word embeddings and wikipedia categories for a low-resource language," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 16, no. 3, p. 18, 2017.

[9] X. Zheng, "Incremental graph-based neural dependency parsing," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1656–1666.

[10] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognitive Computation*, pp. 1–9, 2017.

[11] S. Zhang, K. Duh, and B. Van Durme, "Mt/ie: Cross-lingual open information extraction with neural sequence-to-sequence models," *EACL 2017*, p. 64, 2017.

[12] G. Zhou and J. Huang, "Modeling and learning continuous word embedding with metadata for question retrieval," *IEEE Transactions on Knowledge & Data Engineering*, no. 1, pp. 1–1.

[13] Z. Gan, Y. Pu, R. Henao, C. Li, X. He, and L. Carin, "Learning generic sentence representations using convolutional neural networks," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2380–2390.

[14] J. Rao, H. He, and J. Lin, "Experiments with convolutional neural network models for answer selection," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 1217–1220.

[15] L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, and Y. Yang, "Data-driven answer selection in community qa systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1186–1198, 2017.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[17] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[18] X. Hu, Z. Cai, P. Wiemer-Hastings, A. C. Graesser, and D. S. McNamara, "Strengths, limitations, and extensions of lsa," *The handbook of latent semantic analysis*, pp. 401–426, 2007.

[19] K. Lund and C. Burgess, "Hyperspace analogue to language (hal): A general model semantic representation." in *Brain and Cognition*, vol. 30, no. 3. ACADEMIC PRESS INC JNL-COMP SUBSCRIPTIONS 525 B ST, STE 1900, SAN DIEGO, CA 92101-4495, 1996, pp. 5–5.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[21] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[23] P. D. Turney, "Domain and function: A dual-space model of semantic relations and compositions," *Journal of Artificial Intelligence Research*, vol. 44, pp. 533–585, 2012.

[24] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors." in *ACL (1)*, 2014, pp. 238–247.

[25] T. Schnabel, I. Labutov, D. M. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings." in *EMNLP*, 2015, pp. 298–307.

[26] S. Ghannay, B. Favre, Y. Esteve, and N. Camelin, "Word embedding evaluation and combination." in *LREC*, 2016.

[27] J. Camacho-Collados, M. T. Pilehvar, N. Collier, and R. Navigli, "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017). Vancouver, Canada*, 2017.