Urdu Word Embeddings

Samar Haider
University of Engineering and Technology, Lahore
Pakistan
samar.haider@kics.edu.pk

Abstract

Representing words as vectors which encode their semantic properties is an important component in natural language processing. Recent advances in distributional semantics have led to the rise of neural network-based models that use unsupervised learning to represent words as dense, distributed vectors, called 'word embeddings'. These embeddings have led to breakthroughs in performance in multiple natural language processing applications, and also hold the key to improving natural language processing for low-resource languages by helping machine learning algorithms learn patterns more easily from these richer representations of words, thereby allowing better generalization from less data. In this paper, we train the skip-gram model on more than 140 million Urdu words to create the first large-scale word embeddings for the Urdu language. We analyze the quality of the learned embeddings by looking at the closest neighbours to different words in the vector space and find that they capture a high degree of syntactic and semantic similarity between words. We evaluate this quantitatively by experimenting with different vector dimensionalities and context window sizes and measuring their performance on Urdu translations of standard word similarity tasks. The embeddings are made freely available in order to advance research on Urdu language processing.

Keywords: Urdu, low-resource languages, vector representations, distributional semantics, word embeddings

1. Introduction

Urdu is an Indo-Aryan language that is the national language and lingua franca of Pakistan, and an official language of multiple states of India. There are 109 million speakers of Urdu in Pakistan and 51 million speakers in India. Urdu is also widely spoken across the rest of the world, with more than 163 million total speakers in all countries [1]. Despite its widespread use, both in South Asia and among people of South Asian descent across the world, Urdu remains a low-resource language with few corpora and datasets of appreciable size available for computational tasks. This is not an uncommon phenomenon among low-resource languages, since creation of such resources requires significant time and manpower. However, without sufficient labelled data, it is very difficult to build natural language processing systems that can learn useful patterns which generalize well. This perennial lack of data leads to little research performed on these languages, which in turn leads to few resources created by research.

One way to break out of this loop is to learn higher-level, complex representations of words and phrases that can then be used as input to bootstrap other natural language processing systems downstream. The area of distributional semantics focuses on creating just such representations, wherein semantically similar words are assigned similar representations. Recently, there has been much focus on using neural networks to learn vector representations of words by modelling the task as one of predicting surrounding words from a target word. One advantage of such techniques is that they use unsupervised learning and don't require annotated corpora, which are rare. They can instead be trained on larger, more readily available unannotated corpora, and the learned representations can then be used in natural language processing tasks which use smaller amounts of labelled data.

In this paper, we create the first large-scale distributed vector representations of Urdu words using the skip-gram model introduced by Mikolov et al. (2013a). We collect multiple corpora totalling over 140 million tokens, and train the model on them to learn a vocabulary of more than 100,000 Urdu words. We then take a closer look at the learned representations by looking at relationships between semantically similar words in the vector space. We also evaluate the representations by comparing their performance on word similarity tasks like WordSim-353 and SimLex-999 that test vector space models' ability to learn semantic relations between word pairs. The embeddings are made publicly available online[2] for academic use.

This paper is organized as follows: in Section 2, we present a historical background of distributional semantics, discuss recent advances in the field, and introduce some of its many applications in natural language processing. In Section 3, we describe our own experimental setup, including the corpora used, models trained to learn the word embeddings, and the evaluation tasks and metrics employed. In Section 4, we present our results, both qualitative and quantitative, and discuss the performance of our learned word embeddings on adaptations of standard word similarity tasks. In Section 5, we conclude by summarizing our work and describing our contributions to the improvement of natural language processing for Urdu. We also discuss ideas for future research that builds upon the work done and resources created in this paper.

2. Background

2.1. Distributional Semantics

The crux of the area of distributional semantics is best captured in the words of Firth (1957): "A word is known by the company it keeps." Work on distributional semantics

[1] https://ethnologue.com/language/urd

[2] https://github.com/samarh/urduvec

has thus largely revolved around representing the meaning of a word by the distribution of other words around it.

The 1990s saw the creation of multiple techniques for modelling words in this manner. Church and Hanks (1990) proposed the Pointwise Mutual Information (PMI) measure which sought to quantify the degree of relatedness between two words by looking at how often they occurred together, compared with how often they might occur together if they were not related (independent). Deerwester et al. (1990) invented a model called latent semantic analysis (LSA), which used singular value decomposition (SVD) to create word representations from term-document matrices in an information retrieval setting.

The term 'word embeddings' was first coined by Bengio et al. (2003), who proposed the first neural probabilistic language model. By using a feed-forward neural network with a single hidden layer to predict the next word in a sequence, they laid the foundation of the architecture upon which modern approaches are based. The architecture contained three major building blocks: an embedding layer to generate word embeddings; an intermediate layer to generate an intermediate representation of the words; and a softmax layer to generate a probability distribution over the vocabulary. However, they found that the final softmax layer became the primary bottleneck when training the system due to the cost of computing the function over a large vocabulary.

Collobert and Weston (2008) developed on this work by making a few improvements to the model, the most important of which was the replacement of the expensive cross-entropy criterion with a more efficient pairwise ranking criterion, which greatly improved training speed. They trained word embeddings on a large corpus and showed that the learned embeddings were able to capture the meaning of words quite well, proving useful in higher-level natural language processing tasks (Collobert et al., 2011).

2.2. Skip-gram model

The skip-gram model is one of two neural network architectures introduced by Mikolov et al. (2013a) and later improved upon in (Mikolov et al., 2013b). Together with its sister model, called continuous bag-of-words (CBOW), the two are commonly referred to as 'word2vec'[3], a set of computationally efficient methods for learning vector representations of words.

While the language modelling neural network of Bengio et al. (2003) relied only on past words, the skip-gram model instead included a window of words both before and after the target word when making predictions. This increase in context allowed better predictions and was one of the reasons for improved performance over the language modelling approach. Another improvement was the removal of the computationally expensive hidden layer, in the absence of which the model trained much faster over large corpora.

The two models (skip-gram and CBOW) also differ among themselves in their input and output: while CBOW uses context words to predict the target word, skip-gram does the reverse and uses the target word to predict the context words. A comparison between the performance of different embedding methods by Levy et al. (2015) showed that skip-gram outperforms not only CBOW in the vast majority of cases but also the more recent GloVe method proposed by Pennington et al. (2014).

The training objective of the skip-gram model is to find word embeddings that prove useful for predicting surrounding words, and is defined as:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n\leq j\leq n, \neq 0} \log p(w_{t+j} \mid w_t) \qquad (1)$$

where $p(w_{t+j} \mid w_t)$ is the log probability of a surrounding word given the target word, n is the size of the context window on either side, and T is the size of the corpus.

The softmax output of the skip-gram model is defined as:

$$p(w_{t+j} \mid w_t) = \frac{\exp(v_{w_t}^\top v'_{w_{t+j}})}{\sum_{w_i \in V} \exp(v_{w_t}^\top v'_{w_i})} \qquad (2)$$

where $v_{w_t}^\top v'_{w_{t+j}}$ is the log probability of the surrounding word, which is normalized by sum of the log probabilities of all the words in the vocabulary, V.

Mikolov et al. (2013c) showed that the skip-gram model not only set the state-of-the-art at word similarity tasks, but that the learned embeddings were found to be surprisingly good at capturing both syntactic and semantic relationships and regularities in language.

2.3. Applications

The introduction of the skip-gram model has led to widespread adoption of word embeddings by the natural language processing community. They are now used in a diverse range of applications, some of which we briefly discuss here.

Kim (2014) trained a convolutional neural network on top of pre-trained word embeddings for sentence classification and was able to improve upon previous results with very little parameter tuning. Zou et al. (2013) showed substantial gains in BLEU points at machine translation tasks by using bilingual word embeddings trained from large unlabelled corpora with word alignment constraints to compute semantic similarity of word pairs. Chen and Manning (2014) created a neural network-based dependency parser that used word embeddings as features and found that it showed an improvement in both accuracy and efficiency. dos Santos and Gatti (2014) proposed a method to perform sentiment analysis on short texts using convolutional neural networks with word embeddings as features. Applications of word embeddings have also crossed into other domains: Frome et al. (2013) used convolutional neural networks to predict word embeddings of image labels instead of the labels themselves to exploit semantic information for predicting unseen labels. Vinyals et al. (2015) set the state-of-the-art on multiple image captioning tasks by using convolutional neural networks to embed both images and text in the same vector space for generating image captions.

[3] https://code.google.com/archive/p/word2vec/

3. Experimental Setup

3.1. Corpora

We use three different Urdu corpora to train our model: a corpus with 90 million tokens (Jawaid et al., 2014); a corpus with 35 million tokens (Adeeba et al., 2014); and a dump of the entire Urdu Wikipedia[4]. Extensive preprocessing is performed on all three sources to clean the input. Since our focus is on learning representations of Urdu words only, we remove all non-Arabic script characters from the input. We also remove diacritics in order to normalize words and remove redundant forms. For extracting plain text from the Wikipedia dump, we use Matt Mahoney's script[5] with a few alterations to accommodate Urdu. The post-cleaning statistics of all three corpora, along with their totals, are shown in Table 1.

Corpus	Words	Sentences
(Jawaid et al., 2014)	87,552,394	3,475,529
(Adeeba et al., 2014)	35,347,850	1,429,054
Urdu Wikipedia	17,755,219	527,999
Total	140,655,463	5,432,582

Table 1: Statistics of the corpora used to train the model.

3.2. Model Parameters

We train the skip-gram model implemented in the Gensim toolkit[6] and experiment with varying context window sizes (3, 5, 7) and embedding dimensionalities (100, 200, 300). We pick a minimum frequency cut-off of 10 for a word to be included in the vocabulary and an initial learning rate of 0.025. We use negative sampling to train the model for 5 epochs over the entire text, with the number of noise words to be sampled set to 5.

3.3. Evaluation

For evaluation of the learned word embeddings, we use two benchmark tasks that gauge relationships between different English words: WordSim-353 (Finkelstein et al., 2001) and SimLex-999 (Hill et al., 2015).

WordSim-353 contains 353 word pairs with relatedness scores assigned by 13 to 16 human subjects, and their average used as the final score. SimLex-999 is a more difficult dataset that contains 999 concrete and abstract adjective, noun, and verb pairs. It seeks to measure similarity (cup, mug) rather than relatedness (cup, coffee), and contains similarity scores along with ratings for words' conceptual concreteness assigned to each pair by human subjects. To adapt the two tasks for our work, we translate their word pairs into Urdu using Google's translation service[7].

For comparing our model's predictions with the scores assigned by human subjects, we use the Spearman correlation coefficient (Spearman, 1904), a well-established nonparametric measure of rank correlation between two variables. The rank correlation coefficient, r_s, is defined as:

$$r_s = 1 - \frac{6 \sum_i^n d_i^2}{n(n^2 - 1)} \quad (3)$$

where n is the number of observations and d_i is the difference in rank between the i^{th} observations. Perfect Spearman correlations of $+1$ and -1 occur when the observations of two variables are monotonically increasing or decreasing functions of each other, respectively.

4. Results and Discussion

Training our model on the corpora with the given parameters resulted in a vocabulary of over 100,000 words. This is due to the rich morphology that Urdu exhibits. We then evaluated the accuracy of our word embeddings on the Urdu translations of WordSim-353 and SimLex-999. Due to differences between English and Urdu, a number of English words were either translated into Urdu phrases or left untranslated altogether by the translation service. We thus pruned the datasets to ignore untranslated or out-of-vocabulary (OOV) word pairs that were not found among our learned embeddings. This left us with 269 valid word pairs of WordSim-353 and 691 valid word pairs of SimLex-999.

Table 2 shows the Spearman correlation results of different models on the translation of WordSim-353. We can see that 200- and 300-dimensional embeddings outperform 100-dimensional ones in all cases. The best performing are 200-dimensional embeddings trained with a 5-word context window, achieving a Spearman correlation of 0.524.

		Dimensionality		
		100	200	300
Context	3	0.489	0.516	0.518
	5	0.492	**0.524**	0.513
	7	0.491	0.500	0.510

Table 2: Results of experiments on WordSim-353.

Table 3 shows the Spearman correlation results of different models on the translation of SimLex-999. Here, too, we see a trend of higher dimensional embeddings generally performing better than lower dimensional ones. The best performing embeddings here are 300-dimensional ones trained with a 7-word context window, achieving a Spearman correlation of 0.306.

		Dimensionality		
		100	200	300
Context	3	0.277	0.295	0.294
	5	0.293	0.301	0.301
	7	0.293	0.299	**0.306**

Table 3: Results of experiments on SimLex-999.

Our best performing models achieve Spearman correlations of 0.524 and 0.306 on translations of WordSim-353 and SimLex-999, respectively. For comparison, 300-dimensional embeddings trained by Mikolov et al. (2013a)

[4]https://dumps.wikimedia.org/urwiki/latest/

[5]http://mattmahoney.net/dc/textdata

[6]https://radimrehurek.com/gensim/models/word2vec.html

[7]https://translate.google.com/

using the skip-gram model on 1 billion words of English Wikipedia achieved Spearman correlations of 0.655 and 0.414 on WordSim-353 and Simlex-999, respectively (Hill et al., 2015). This shows that, despite challenges in translating English words accurately into Urdu, our embeddings have captured semantic relationships between words quite well.

To take a closer look at the kind of semantic relationships captured in the word embeddings, we find the ten closest vectors to a given word in the vector space using the cosine similarity. Figure 1 shows three examples of this, along with English translations, for the Urdu words for 'Lahore' (city), 'room', and 'car', respectively. Inspecting the results, it is clear to see that the embeddings have captured very meaningful syntactic and semantic relationships between words. They have also captured semantic similarity between different spelling variations and morphological forms found in Urdu.

(Lahore)	**لاہور**	**(room)***	**کمرہ**	**(car)**	**گاڑی**
(Karachi)	کراچی	(rooms)†	کمرے	(motor)	موٹر
(Rawalpindi)	راولپنڈی	(room)*	کمرا	(jeep)	جیپ
(Gujranwala)	گوجرانوالہ	(flat)	فلیٹ	(cycle)	سائیکل
(Multan)	ملتان	(veranda)	برآمدہ	(motorcycle)	موٹر سائیکل
(Peshawar)	پشاور	(bedrooms)	بیڈروم	(wagon)	ویگن
(Sialkot)	سیالکوٹ	(rooms)†	کمروں	(bicycle)	بائیسکل
(Amritsar)	امرتسر	(verandas)	برآمدے	(train)	ٹرین
(Sheikhupura)	شیخوپورہ	(hall)	ہال	(rickshaw)	رکشہ
(Sargodha)	سرگودھا	(cabin)	کیبن	(bike)	بائیک
(Hyderabad)	حیدرآباد	(hostel)	ہاسٹل	(cars)	گاڑیوں

Figure 1: Words most similar (in descending order of similarity) to those in bold. An * represents spelling variations and a † represents different morphological forms of the same word.

5. Conclusion

The introduction of computationally efficient neural network-based methods for unsupervised learning of word embeddings from large unannotated corpora has been a watershed moment in natural language processing in recent years. The use of embeddings has not only improved the state-of-the-art in multiple natural language processing applications, but has also provided an impetus to research on low-resource languages. In this paper we presented work done on creating the first large-scale word embeddings for Urdu, a low-resource albeit widely-spoken South Asian language that has a large population of native speakers in Pakistan and India. We performed quantitative evaluation of the embeddings by adapting standard word similarity tasks to Urdu, and investigated relationships between the learned embeddings by looking at words predicted by the model to be semantically similar.

In the future, we plan on refining these embeddings further as well as creating custom benchmark tasks for evaluating them, keeping in mind the distinct characteristics of the Urdu language. We also plan on performing extrinsic evaluation of these embeddings by using them in tasks like text classification, named entity recognition, sentiment analysis, dependency parsing, and machine translation. We are optimistic of this direction and strongly believe that this resource will play a major role in improving natural language processing for Urdu.

6. Bibliographical References

Adeeba, F., Akram, Q., Khalid, H., and Hussain, S. (2014). CLE urdu books n-grams. In *Conference on Language and Technology, CLT 14, Karachi, Pakistan*.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar*, pages 740–750.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning, (ICML 2008), Helsinki, Finland*, pages 160–167.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics, COLING 2014, Dublin, Ireland*, pages 69–78.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: the concept revisited. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China*, pages 406–414.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, USA*, pages 2121–2129.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Confer-*

ence on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, pages 1746–1751.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, USA*, pages 3111–3119.

Mikolov, T., Yih, W., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar*, pages 1532–1543.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, Massachusetts, USA*, pages 3156–3164.

Zou, W. Y., Socher, R., Cer, D. M., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Seattle, Washington, USA*, pages 1393–1398.

7. Language Resource References

Jawaid, Bushra and Kamran, Amir and Bojar, Ondřej. (2014). *Urdu Monolingual Corpus*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.