Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

Towards Bengali Word Embedding: Corpus Creation, Intrinsic and Extrinsic Evaluations

Md. Rajib Hossain and Mohammed Moshiul Hoque
Department of Computer Science & Engineering, Chittagong University of Engineering and Technology, Chittagong-4349 Bangladesh
rajcsecuet@gmail.com, moshiul_240@cuet.ac.bd

Abstract

Distributional word vector representation or word embedding has become an essential ingredient in many natural language processing (NLP) tasks such as machine translation, document classification, information retrieval and question answering. Investigation of embedding model helps to reduce the feature space and improves textual semantic as well as syntactic relations. This paper presents three embedding techniques (such as Word2Vec, GloVe, and FastText) with different hyperparameters implemented on a Bengali corpus consists of 180 million words. The performance of the embedding techniques is evaluated with extrinsic and intrinsic ways. Extrinsic performance evaluated by text classification, which achieved a maximum of 96.48% accuracy. Intrinsic performance evaluated by word similarity (e.g., semantic, syntactic and relatedness) and analogy tasks. The maximum Pearson ($\hat{r}$) correlation accuracy of 60.66% ($S_{s\hat{r}}$) achieved for semantic similarities and 71.64% ($S_{y\hat{r}}$) for syntactic similarities whereas the relatedness obtained 79.80% ($R_{s\hat{r}}$). The semantic word analogy tasks achieved 44.00% of accuracy while syntactic word analogy tasks obtained 36.00%.

1 Introduction

Word embedding is a distributional vector representation of words in which syntactic and semantic interpretations are derived from the enormous amount of unlabeled texts. Recently, the word embedding is considered as a powerful tool due to its many applications in NLP, thus, gained much attention by NLP experts. This is a growing up research issue for well-resourced language like English, where an embedding algorithm generates a model (Devlin et al., 2019). However, it is a very complicated task to adopt an embedding algorithm (of any language) directly for the resource-constrained languages such as Bengali due to the scarcity of resources. As a result, the low-resource language is trail end in NLP tools development. Bengali is the most popularly communicated language of Bangladesh while second most communicated of the 22 official languages of India which makes Bengali is the 7^{th} most spoken language in the world (Hossain and Hoque, 2019). However, due to the shortage of resources, the development of NLP tools are striving. Bengali speaking people are suffering to access modern NLP tools that might be affected their sustainable use of language technologies. Therefore, Bengali word embedding is an essential prerequisite to developing any Bengali language based NLP tools.

There are two well-known evaluation methods used extensively to evaluate embedding techniques such as extrinsic and intrinsic (Zhelezniak et al., 2019a). Extrinsic evaluation refers to downstream tasks like as machine translation (MT) (Banik et al., 2020), and Part of speech (POS) tagging (Priyadarshi and Saha, 2020). Intrinsic evaluation goals to evaluate the quality of language processing tasks, such as semantic and syntactic word similarity (Pawar and Mago, 2018), Word relatedness (Gladkova and Drozd, 2016), and Word analogy (Schluter, 2018). Unavailability of standard Bengali embedding corpus and inadequacy of resources are antecedents that make such a model generation and evaluation very challenging. Moreover, there is no generalized embedding model available to date for Bengali downstream tasks. Thus, the proposed work introduces a Bengali embedding model generation and evaluation techniques with different hyperparameters settings. Specifically, the key contributions of this work are:

- Acquire raw monolingual Bengali corpora with 180 million words where the unique words are 13 million.
- Construct and annotates the intrinsic and extrinsic evaluation datasets as well as evaluate the annotation purity.

© 2020 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

- Generate *ninety* embedding models with the combination of three different algorithms (such as Word2Vec, GloVe,and FastText) and variations of model parameters.
- Examine the influence of hyperparameters on the embedding models performance.

As far as we are aware, the proposed work is the first attempt to generate large-scale embedding models evaluates with intrinsic and extrinsic evaluators.

2 Related Work

Distributional word vector representation or embedding model generation is a well-established research agenda in NLP domain. There are plenty of research works have been carried out on word embedding in high-resource languages, but it remains as a barrier for low resource languages. The first intrinsic evaluation datasets introduced in RG-65 (Rubenstein and Goodenough, 1965) which contains 65 contextual synonymy pairs. The WordSimilarity-353 introduced by (Finkelstein et al., 2001) and the dataset contains 353 words pair with 13 different subjects. In recent time, three embedding model evaluation datasets have been introduced: SimLex-999 (Hill et al., 2015), SemEval 2017 (Camacho-Collados et al., 2017), and MEN (Bruni et al., 2014). An Italian language embedding model has developed in (Di Gennaro et al., 2020), which achieved 53.74% overall analogies accuracy for $19,791$ texts. Moreover, this work achieved a semantic analogies accuracy of 59.20% for $8,915$ texts and syntactic accuracy of 48.80% for $10,876$ texts. However, this work considered only 3COSADD similarity score and not consider the word relatedness and extrinsic evaluations. Ercan and Yıldız (2018) devised a Turkish word similarity and relatedness system that produced Turkish embedding dataset derived from English word similarity and relatedness datasets (e. g. WordSimilarity-353, MEN, and SimLex-999). This work achieved spearman score (ρ) of 0.667 for WordSimilarity-353, 0.68 for MEN and 0.67 for SimLex-999 respectively. The developed embedding model was not evaluated with extrinsic evaluation (Chiu et al., 2016).

Although most of the current works on embedding model, resource creation and evaluations conducted for the high-resource languages (e.g., English, Germany, and French) there are few comprehensive research conducted on low resource languages such as Assamese, Gujarati, Hindi, Kannada, and so on (Kumar et al., 2020) and Turkish (Ercan and Yıldız, 2018). Kumar et al. (2020) introduced the pre-trained word embedding models for Indian languages. The proposed system generates 14 Indian local language embeddings with 8 different approaches, including 436 models. Embedding models are evaluated by extrinsic evaluators and archived more than 90.00% accuracy for UPOS, and XPOS tagging using universal dependency treebank datasets (Nivre et al., 2016). The NER tagging accuracy about 95.00% with FastText embedding. Kumar et al. (2020) aimed to solve 14 Indian languages NLP problems, but the generated models are not considered intrinsic evaluations. To best of our knowledge, only single research was conducted concerning Bengali word embedding using Word2Vec (Sadman et al., 2019). However, this work considered only intrinsic evaluations with a self-build dataset. Our approach considered *ninety* embedding models based on GloVe, FastText and Word2Vec models and measured the performance using intrinsic and extrinsic evaluators.

3 Methodology

The principal aim of our research is to investigate the affect of intrinsic and extrinsic evaluations on Bengali word embedding models. Thus, the proposed scheme comprises of three main parts: corpus creation, word embedding model development, and evaluation. Figure 1 illustrates the abstract view of our work.

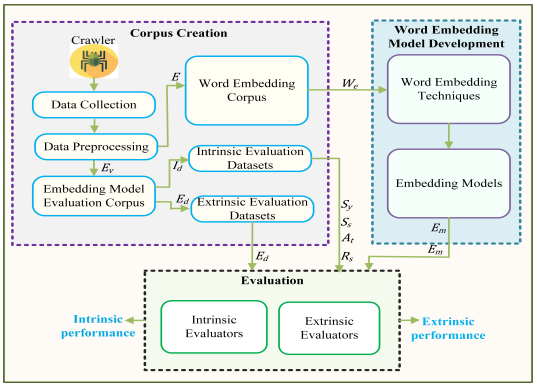

Figure 1: Abstract process of evaluations for Bengali word embedding.

3.1 Corpus Creation

We collected Bengali texts from various online sources and distributed these into two sets: word

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

embedding corpus (E) and embedding model evaluation corpus (E_v). We used a Python crawler to crawl data. We collected $910,720$ Bengali text files over a twenty-four month period (September 10, 2018, to September 11, 2020) which are forwarded to data preprocessing step. Initially, the non-Bengali alphabets and digits are removed from the text files. In the next, preprocessing step removed HTML tags, hashtags, URLs, punctuation and white spaces. Finally, the duplicate texts are deleted from the archive. The preprocessing step produced of $882,352$ usable text, and removed $28,368$ blank size text documents from the initial dataset due to various preprocessing operations. These usable preprocessed data are randomly distributed into two sets; one set for embedding model evaluation ($100,000$ texts) and another set for word embedding corpus ($782,352$ texts). The embedding corpus (W_e) (i.e., total $180,081,093$ words) is fed to the embedding techniques.

Embedding model evaluation corpus (E_v) is a combination of intrinsic (I_d) and extrinsic datasets (E_d). In order to perform an extrinsic evaluation, out of $100,000$ text documents, a total of $60,000$ documents are randomly selected. This dataset was labelled manually followed by majority voting to assign the suitable label. Two linguistic experts are assigned to annotated each data into one of the six pre-defined categories such as Accident (A_t), Crime (C_e), Entertainment (E_t), Health (H_h), Politics (P_s) and, Sports (S_p). Among $60,000$ text documents, both experts have been agreed upon $54,858$ text labels. The developed corpus (E_d) achieved a Kappa score (K) is 78.53%, which indicates a reasonable agreement between annotators for downstream task. To perform intrinsic evaluations of the embedding models four different sub datasets are used for conducting four measures: semantic word similarity (S_s), syntactic word similarity (S_y), relatedness (R_r), and analogy tasks (A_t). Intrinsic datasets and corresponding kappa score are shown in Table 1. Here, S_s, S_y and R_s datasets are substantial agreement where as A_t is moderate agreement.

Datasets	No. of samples	Kappa score (%)
S_s	100	71.55
S_y	100	65.30
R_s	100	75.13
A_t	100	56.90

Table 1: Embedding model datasets and kappa score.

3.2 Word Embedding Model Development

We consider three well-known embedding techniques for Bengali corpus including Word2Vec, GloVe, and FastText. To realize the effect of hyperparameters, we have considered embedding dimension ($size$), minimum word frequency count (min_count), contextual windows size ($window$) and number of iteration ($epoch$) for each of the embedding technique.

Word2Vec (skip-gram and CBOW): Word2Vec (Mikolov et al., 2013a) technique takes W_e as the input and produce the embedding model E_m using gensim library (Řehůřek and Sojka, 2010). Two versions of Word2Vec pre-trained models such as Skip-gram and continuous bag of words (CBOW) are used with similar hyperparameters for tuning: $size$: $\{50, 100, 150, 200, 250, 300\}$, $window$: $\{5, 10, 15\}$, min_count : $\{2\}$ and $epochs$: 30 respectively. There are 36 embedding models (e.g., 18 for CBOW and 18 for skip-gram) generated for W_e using Word2Vec technique.

GloVe: GloVe (Pennington et al., 2014) technique generated 18 embedding models (E_m) for the embedding corpus W_e. Different hyperparameters are used to optimize the model such as $size$: $\{50, 100, 150, 200, 250, 300\}$, min_count : $\{2\}$, X_MAX : 95, $epochs$: 30 and $window$: $\{5, 10, 15\}$ respectively. The remaining hyperparameters remains the same as default settings. Finally, the eighteen embedding models (E_m) are evaluated by the evaluators.

FastText (skip-gram and CBOW): FastText embedding technique (Bojanowski et al., 2017), takes W_e as input and generates a embedding model (E_m) as output using gensim library (Řehůřek and Sojka, 2010). Different values of hyperparameters are used to achieve optimize performance such as $size$: $\{50, 100, 150, 200, 250, 300\}$, $window$: $\{5, 10, 15\}$, min_count : $\{2\}$ and $epochs$: 30 respectively. Our approach produced of 36 models (e.g., 18 for FastText-skip-gram and 18 for FastText-CBOW) that are evaluated using evaluators.

4 Results and Discussion

Intrinsic evaluations are performed for a total of *ninenty* (e.g., Word2Vec=36, GloVe=18 and Fast-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

			Semantic (%)		Syntactic (%)		Relatedness (%)	
Model	size	window	$S_{s\hat{\rho}}$	$S_{s\hat{r}}$	$S_{y\hat{\rho}}$	$S_{y\hat{r}}$	$R_{s\hat{\rho}}$	$R_{y\hat{r}}$
GloVe	**300**	**15**	**60.02**	**60.66**	69.54	70.62	79.20	79.72
	250	**10**	56.33	57.77	**70.41**	70.66	**79.22**	**79.80**
	250	**5**	56.93	57.90	69.87	**71.64**	79.20	78.33
FastText (SG)	250	10	53.64	53.38	39.47	38.31	68.28	67.54
	150	10	56.78	55.18	37.78	36.03	66.63	65.84
Word2Vec (SG)	250	5	44.31	45.10	31.78	30.51	56.43	57.41
FastText (Grave et al., 2018)	300	-	49.41	49.78	5.93	-1.55	47.23	43.91

Table 2: Performance of embedding models concerning semantic, syntactic and relatedness word similarity.

Text=36) embedding models. Among these, the results of best *four* embedding models presented for extrinsic and intrinsic evaluators.

4.1 Intrinsic evaluation results

The word similarity (semantic and syntactic) score is calculated by Cosine similarity (C). The model performance can be calculated from the spearman ($\hat{\rho}$) and pearson ($\hat{r}$) correlations (Zhelezniak et al., 2019b). The well-known word analogy solver, 3COSADD (Mikolov et al., 2013b) is used to solve the analogy tasks. Three similarity measures are used to evaluate word similarity and analogy tasks analysis. In order to maintain consistency, we performed training for all models with our developed corpus.

Word similarity: Table 2 shows the intrinsic evaluations performance of the embedding models. Annotators word similarity rates are range from $0-10$ whereas the cosine similarity score normalized by ten times. All values in Table 2 are normalized by hundreds times. Maximum semantic correlation values are $S_{s\hat{\rho}} = 60.02\%$ and $S_{s\hat{r}} = 60.66\%$ for GloVe ($size = 300$ and $window = 15$) technique. Highest syntactic correlation is $S_{y\hat{\rho}} = 70.41\%$ for GloVe ($size = 250$ and $window = 10$) where as $S_{y\hat{r}} = 71.64\%$ for GloVe ($size = 250$ and $window = 5$) techniques. The R_s highest correlations, $R_{s\hat{\rho}=79.22\%}$ and $R_{y\hat{r}=79.80\%}$ have been achieved using GloVe ($size = 250$ and $window = 10$) technique. There are *eight* semantic words pair are not able to process by E_m where as *four* syntactic words pair are not able to process by E_m of all techniques. Relatedness words pair are fully processed by all embedding techniques.

Word analogy results: The semantic analogy results are shown in Table 3, while Table 4 denotes the syntactic analogy tasks performance based on our corpus. Due to unavailability of Bengali semantic and syntactic analogy datasets, we have been developed A_t datasets were 50 analogy words used for semantic and another fifty used to perform syntactic analogy tasks. GloVe ($size = 300$ and $window = 15$) technique has achieved maximum accuracy of 38.00% (Add) and 44.00% (Mull) for semantic analogy tasks. Minimum semantic analogy tasks accuracy is obtained by FastText (Grave et al., 2018) embedding model. The maximum

Semantic analogy tasks accuracy (%)				
Model	size	window	Add	Mull
GloVe	**300**	**15**	**38.00**	**44.00**
FastText (SG)	250	10	30.00	34.00
Word2Vec(SG)	250	5	26.00	30.00
FastText (Grave et al., 2018)	300	-	20.00	26.00

Table 3: Analogy tasks performance summary for semantic datasets.

syntactic analogy tasks accuracy are 30.00% (Add) and 36.00% achieved by GloVe ($size = 300$ and $window = 15$) E_m, while 20.00% (Add) and 24.00% (Mull) from FastText (Grave et al., 2018) embedding model.

4.2 Extrinsic evaluation results

The E_d is a Bengali text classification dataset which partitioned into the three sets: training $(39,079)$, validation $(6,000)$ and testing $(9,779)$. The text classifier model trained with a multi-kernel CNN architecture (Kim, 2014). The performance of the text classifier model assesses with extrinsic

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

Syntactic analogy tasks accuracy (%)				
Model	size	window	Add	Mull
GloVe	**300**	**15**	**30.00**	**36.00**
FastText (SG)	250	10	26.00	28.00
Word2Vec(SG)	250	5	20.00	26.00
FastText (Grave et al., 2018)	300	-	20.00	24.00

Table 4: Analogy tasks performance for syntactic datasets.

evaluators including accuracy (A), micro average F1-score, average precision (A_p), average recall (A_r) and confusion matrix (CM) (Wu et al., 2020). The evaluators evaluated the embedding models (E_m) downstream task (e.g., text classification) performance (in Tables 5 and 6). Table 5 shows the summary of text classification performance.

Models	size/window	F1-score(%)	A(%)
Word2Vec	250/10	93.43	93.87
GloVe	**200/10**	**96.03**	**96.48**
FastText	200/15	95.57	95.71

Table 5: Extrinsic evaluation for text classification.

The GloVe model achieved the highest accuracy of 96.48%. For clarity, we presented only the results of best four embedding models out of ninety models. Table 6 depicts the confusion matrix of GloVe model ($size = 200$ and $window = 10$) for text classification performance. The maximum correctly predicted class is *Politics* and incorrectly predicted class is *Crime*. The highest misclassification occurred for *Crime* and *Accident* pair.

CM	A_t	C_e	E_t	H_h	P_s	S_p
A_t	1636	43	1	3	3	2
C_e	62	1468	2	10	27	3
E_t	1	4	1603	12	8	16
H_h	1	5	16	1593	12	9
P_s	3	15	3	11	1570	6
S_p	1	6	43	4	12	1565

Table 6: Confusion matrix of text classification task.

Figure 2 shows few example scores for semantic, syntactic and relatedness words pair score obtained from GloVe model and human annotators. GloVe and FastText (SG) models accuracy are considerable for semantic and relatedness similarities. In the case of extrinsic evaluations, the performance

	$Word_1$	$Word_2$	Annotators score (Avg.)	Cosine Similarity (C)
Semantic Words pair	অপরাধ Aparādha Crime	পাপ pāpa Sin	4.00	3.10
Semantic Words pair	জল Jala Water	পানি Pāni Water	8.80	6.69
Relatedness Words pair	স্কুল Skula School	কলেজ kalēja College	8.50	7.62
Relatedness Words pair	শার্ট Śārṭa Shirt	প্যান্ট Pyānṭa Pants	8.70	8.32
Syntactic Words pair	শিক্ষক Śikṣaka Teacher	শিক্ষকরা Śikṣakarā Pants	7.3	6.60
Syntactic Words pair	পর্বত Parbata Mountain	পর্বতমালা Parbatamālā Mountains	4.30	3.25

Figure 2: Word pair similarity scores, the Cosine similarity score is normalized by 10 times and annotators score is ranging between 1 to 10.

of GloVe and FastText embedding models are significant for the text classification task.

5 Conclusion and Future Work

In this work, we have been generated about *ninety* embedding models for the Bengali language. These models have developed using the combinations of three embedding techniques (such as GloVe, Word2Vec, and FastText) and various hyperparameters. All models have evaluated by extrinsic and intrinsic evaluators on our developed corpus. The performance of an embedding model significantly depends on the hyperparameters, corpus and nature of the model. Although GloVe model performed better than Word2Vec and FastText, there is no generalized embedding model for intrinsic and extrinsic NLP tasks. The embedding models are highly corpus oriented, and hyperparameters also vary from one task to another. In the future, the existing Bengali corpus can be extended for embedding model generation to alleviate the out-of-vocabularies problems. The context-dependent feature represents technique (such as BERT, ElMo and XLNet) will be investigated to find suitable embedding technique for Bengali. In addition to that, more analogy tasks can be considered to assess the performance of different embedding models with various intrinsic and extrinsic evaluators.

References

Debajyoti Banik, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Statistical machine translation based on weighted syntax–semantics. *Sādhanā*, 45:1–12.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver, Canada. Association for Computational Linguistics.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Giovanni Di Gennaro, Amedeo Buonanno, Antonio Di Girolamo, Armando Ospedale, Francesco A. N. Palmieri, and Gianfranco Fedele. 2020. An analysis of word2vec for the italian language. *Smart Innovation, Systems and Technologies*, pages 137–146.

Gökhan Ercan and Olcay Taner Yıldız. 2018. AnlamVer: Semantic model evaluation dataset for Turkish - word similarity and relatedness. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3819–3836, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, page 406–414, New York, NY, USA. Association for Computing Machinery.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42, Berlin, Germany. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Md. Rajib Hossain and Mohammed Moshiul Hoque. 2019. Automatic bengali document categorization based on deep convolution nets. In *Emerging Research in Computing, Information, Communication and Applications*, pages 513–525, Singapore. Springer Singapore.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Saurav Kumar, Saunack Kumar, Diptesh Kanojia, and Pushpak Bhattacharyya. 2020. "a passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 352–357, Marseille, France. European Language Resources association.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Atish Pawar and Vijay Mago. 2018. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *CoRR*, abs/1802.05667.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 December 2020 doi:10.20944/preprints202012.0600.v1

on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ankur Priyadarshi and Sujan Kumar Saha. 2020. Towards the first maithili part of speech tagger: Resource creation and system development. *Computer Speech Language*, 62:101054.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.

Nafiz Sadman, Akib Sadmanee, Md. Iftekhar Tanveer, Md. Ashraful Amin, and Amin Ahsan Ali. 2019. Intrinsic evaluation of bangla word embeddings. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

Natalie Schluter. 2018. The word analogy testing caveat. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 242–246, New Orleans, Louisiana. Association for Computational Linguistics.

Di Wu, Mengtian Zhang, Chao Shen, Zhuyun Huang, and Mingxing Gu. 2020. Btm and glove similarity linear fusion-based short text clustering algorithm for microblog hot topic discovery. volume 8, pages 32215–32225. IEEE Access.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Hammerla. 2019a. Correlation coefficients and semantic textual similarity. 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 951–962, Minneapolis, Minnesota. Association for Computational Linguistics.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Hammerla. 2019b. Correlation coefficients and semantic textual similarity. 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 951–962, Minneapolis, Minnesota. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50, Valletta, Malta. University of Malta.