Geographical Evaluation of Word Embeddings

Michal Konkol[1], **Tomáš Brychcín**[1], **Michal Nykl**[1], and **Tomáš Hercig**[1,2]

[1]NTIS – New Technologies for the Information Society,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
[2]Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
{konkol,brychcin,nyklm,hercig}@kiv.zcu.cz

Abstract

Word embeddings are commonly compared either with human-annotated word similarities or through improvements in natural language processing tasks. We propose a novel principle which compares the information from word embeddings with reality. We implement this principle by comparing the information in the word embeddings with geographical positions of cities. Our evaluation linearly transforms the semantic space to optimally fit the real positions of cities and measures the deviation between the position given by word embeddings and the real position. A set of well-known word embeddings with state-of-the-art results were evaluated. We also introduce a visualization that helps with error analysis.

1 Introduction

In recent years the improvements in quality of word embeddings led to significant improvements in many natural language processing (NLP) tasks, e.g. sentiment analysis (Maas et al., 2011), named entity recognition (Lample et al., 2016), or machine translation (Zou et al., 2013). New models for word embeddings and improvements to the old ones are introduced rapidly (Bojanowski et al., 2017; Salle et al., 2016; Yin and Schütze, 2016). As the number of various word embeddings increases, it becomes very time consuming to choose word embeddings for a particular task (Nayak et al., 2016).

To mitigate the problem, it is necessary to provide appropriate evaluation together with the word embeddings. The evaluation should cover multiple properties of word embeddings in order to allow the user to choose the model directly based on the results (Nayak et al., 2016). Many evaluation approaches have already been proposed and they can be roughly divided to intrinsic and extrinsic (Schnabel et al., 2015).

The intrinsic evaluation measures the quality of the model directly by comparison with human-annotated data that capture semantic information. The advantage of this approach is that it is fast, simple, and easy to reproduce and analyze (Schnabel et al., 2015; Nayak et al., 2016). The main issue is that the evaluation score often does not correlate with improvements in NLP tasks (Chiu et al., 2016).

The extrinsic evaluation is indirect and measures the improvements through other tasks – currently mainly through NLP tasks. The advantage of this approach is that for each task we know which model to choose. The disadvantage is the computational complexity (Nayak et al., 2016). For each new word embeddings we need to train models for several approaches to several tasks and find the optimal hyperparameters of the models. Moreover, the same data and implementations should be used by all researchers for the evaluation.

We propose a new evaluation paradigm that is in between the intrinsic and extrinsic evaluation (actually, half the people believe its intrinsic and the other half believe its extrinsic). We measure neither the semantic word similarity as in intrinsic evaluation nor improvements in a particular task that uses word embeddings. We compare the information encoded in word embeddings directly with real-world data. We implement the paradigm with geographical data. We take GPS coordinates of cities and measure to what degree is the information encoded in the word embeddings.

The paper is organized as follows. In Section 2 we describe commonly used evaluation approaches for word embeddings and discuss their

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 224–232,
Taipei, Taiwan, November 27 – December 1, 2017

strengths and weaknesses. Our evaluation metric is introduced in Section 3. In Section 4 we provide various experiments with our evaluation metric, including evaluation of state-of-the-art word embeddings. Finally, we conclude in Section 5.

2 Related Work

There are two common tasks which fall under intrinsic evaluation: word similarity and word analogy tasks.

In the word similarity task, the evaluation data consist of pairs of words and their similarity annotated by humans. The word embeddings are compared with the evaluation data usually by Spearman rank correlation. The word similarity task has a long tradition in the semantics research (Rubenstein and Goodenough, 1965). Currently there are multiple corpora created to test different properties of the word embeddings (Finkelstein et al., 2001; Agirre et al., 2009; Luong et al., 2013; Hill et al., 2015).

The word analogy task evaluates the ability of the word embeddings to capture relations between words consistently. The evaluation data consists of questions (with answers) in the form: if word a is related to word b the same way as word c is related to word d, what word is d given a, b, and c? The word embeddings are compared based on their accuracy. The Google Word Analogy corpus is usually used for evaluation (Mikolov et al., 2013a). The word analogy task is closest to our evaluation because some of the questions are also based on real-world data, e.g. countries and their capital cities. Unlike our evaluation, they handle city names as common words, use the global semantic space, and compare them using cosine similarity.

The extrinsic evaluation uses other NLP tasks for comparison of word embeddings. Many tasks are used for extrinsic evaluation, e.g. sentiment analysis (Schnabel et al., 2015), named entity recognition (Konkol et al., 2015), or parsing (Bansal et al., 2014). Word embeddings are compared based on the improvements measured with standard evaluation metrics for the given task.

Both intrinsic and extrinsic evaluations have their advantages and disadvantages. The word similarity task was analysed and criticized by multiple authors (Faruqui et al., 2016; Chiu et al., 2016; Batchkarov et al., 2016; Gladkova and Drozd, 2016). The advantages of word similarity evaluation are that it is very fast and can be easily interpreted from the linguistic point of view (or generally by human). The corpora often suffer from a subset of the following disadvantages: low correlation with extrinsic evaluation (applications), polysemy is not supported, subjectivity of single value similarity, overfitting (no training, heldout, test sets), significance tests are not common for word similarity, and the data are often small.

The word analogy task has the same disadvantages as the word similarity task; moreover the evaluation is quite slow, because it is necessary to sort all words based on their similarity with the question. Linzen (2016) provides a detailed analysis of the word analogy task and shows that results in this evaluation are to a large extent based on proximity in the semantic space rather than consistent offsets between the word pairs.

The main advantage of the extrinsic evaluation is that it directly measures application improvements. The main disadvantage is computational complexity. There exist many tasks that could be used for evaluation, but it is intractable to use all of them (Nayak et al., 2016). Moreover, there exist many approaches to all the tasks and some embeddings might be good for one approach and bad for the others. Choosing a single approach as a general benchmark could lead to incorrect conclusions. If we still want to choose a single model, then which one? On one hand, the state-of-the-art approaches of the tasks evolve in time – state-of-the-art method may well become a baseline in a few years. On the other hand, using baseline approaches loses the ability to measure application improvements. Word embeddings may have a high score with the baseline approach, but may contain the same information that is already present in the state-of-the-art approach. Other embeddings may have low score with the baseline approach, but the information may be usable in the state-of-the-art approach.

Many disadvantages of intrinsic evaluation are also related to particular tasks in extrinsic evaluation, e.g. named entity recognition or sentiment analysis usually do not use significance tests, are subjective, or use small data sets.

Nayak et al. (2016) propose a system for standard automatic extrinsic evaluation. They selected a representative subset of tasks for the evaluation and chose a single approach for each task (based

on standard neural network architectures) in order to achieve reasonable evaluation times (4-5 hours). Even though this approach has the disadvantages presented in the previous two paragraphs, it is definitely a step forward to a standardized evaluation.

3 Proposed Evaluation

The evaluation data set consists of a list of n names of cities and their GPS coordinates stored in matrix $\mathbf{G} \in \mathbb{R}^{n \times 2}$. We assume that Earth is perfectly spherical and its radius is 6,371. Given the assumption, the GPS coordinates in matrix $\mathbf{G}$ can be transformed to Euclidean coordinates in matrix $\mathbf{Y} \in \mathbb{R}^{n \times 3}$ and back. The word embeddings of cities are in matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with a d-dimensional vector for each city name. We normalize rows of $\mathbf{X}$ and $\mathbf{Y}$, because it is helpful for stability of the optimization and we need only the cosine similarity between the rows.

The first step of our evaluation is to find a subspace of the original d-dimensional word embeddings space that contains the information about city locations. The word embeddings transformed to the subspace are represented by matrix $\mathbf{W} \in \mathbb{R}^{n \times 3}$. The matrices $\mathbf{W}$ and $\mathbf{Y}$ have to share the same dimensions because we want to compare the distances between their rows (cities). We are looking for a linear transformation $\mathbf{W} = \mathbf{XT}$ parametrized by transformation matrix $\mathbf{T} \in \mathbb{R}^{d \times 3}$. We use the least squares cost function, the optimal transformation matrix $\mathbf{T}^*$ is defined as a transformation matrix that minimizes squared distances between real and approximate city positions $\|\mathbf{W} - \mathbf{Y}\|_2$. This optimization problem is highly prone to overfitting as $n \approx d$; moreover the row rank of $\mathbf{X}$ is likely lower than n, because the embeddings for cities are highly correlated and thus they are likely linearly dependent. Thus we employ $\boldsymbol{L_2}$ regularization. The final optimization problem is given by Equation (1), where α is the regularization weight.

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} (\|\mathbf{XT} - \mathbf{Y}\|_2 + \alpha \|\mathbf{T}\|_2) \quad (1)$$

Finally, we can compare $\mathbf{W}$ and $\mathbf{Y}$. The primary metric for the evaluation is mean geographic distance, i.e. the distance between two points on a globe measured on the surface. We firstly need to normalize rows of $\mathbf{W}$ because the vectors can be above or below the surface. The geographic distance can be measured using Equation (2), where g is the geographic distance, $\mathbf{w}_i$ and $\mathbf{y}_i$ denote the i-th row of $\mathbf{W}$ and $\mathbf{Y}$ respectively, and r is the radius of Earth.

$$g = r \cdot \arccos(\mathbf{w}_i \cdot \mathbf{y}_i) \quad (2)$$

While the mean geographic distance is a good metric for a global view, it does not take the local structure into account, i.e. a random model moves the cities in all directions and breaks the local structure (nearest neighbors), but other model (with the same mean geographic distance) can move the cities in one direction and preserve local structure. We measure the ability of the embeddings to capture local structure by Precision at K (Prec@K). This metric creates two sets of K nearest neighbors for each city, one for the evaluation data $\mathbf{Y}$ and one for the transformed word embeddings $\mathbf{W}$. The precision between these two sets is averaged over all cities.

We also provide more statistics that help with understanding of the primary score. Median geographic distance gives a better idea about common distances, because it is not affected by extreme values. Sometimes, we found it easier to think about the errors in angles rather than distances, mainly because angles are independent of the size of the globe.

4 Experiments

In this section we firstly describe the data used for the proposed evaluation. Then we briefly introduce the word embeddings used to demonstrate the proposed evaluation. Finally, we follow with experiments that show some properties of the evaluation.

4.1 Data

We downloaded the list of 640 known cities from `https://www.timeanddate.com/worldclock/full.html` and further adjusted it. We removed cities that consist of multiple words from the list, because the evaluated models were trained only on single word expressions. It has lead to a reduction of the set to 540 cities. Then we created a dictionary of the top 10,000 words from Wikipedia and filtered out cities not present in the models, which resulted into a set of 483 cities. Finally we removed cities with ambiguous names and inconsistent use of diacritics, leaving us with 440 cities.

Model	Data	Dimension	Mean distance	Median distance	Mean angle	Prec@10	Prec@20
Random placement	—	—	10007	10007	90°	0.03	0.06
Random embeddings	—	300	10040	10422	90.3°	0.031	0.065
GloVe	6B	50	3776	3086	34.0°	0.144	0.236
GloVe	6B	100	3177	2565	28.6°	0.150	0.258
GloVe	6B	200	2756	2158	24.8°	0.210	0.336
GloVe	6B	300	2604	2116	23.4°	0.218	0.339
GloVe	42B	300	2504	1948	22.5°	0.192	0.313
GloVe	840B	300	2044	1681	18.4°	0.260	0.408
LexVec – cc	58B	300	1992	1662	17.9°	0.267	0.414
LexVec – w + nc	7B	300	1908	1508	17.2°	0.304	0.439
MetaEmbeddings	—	200	3322	2845	29.9°	0.129	0.237
SkipGram – BoW2	1-5B	300	2279	1762	20.5°	0.278	0.407
SkipGram – BoW5	1-5B	300	1985	1642	17.9°	0.273	0.422
SkipGram – Dep	1-5B	300	3240	2464	29.1°	0.176	0.265
FastText	1-5B	300	1686	1429	15.2°	0.338	0.482
WoRel	2.5B	300	1921	1487	17.3°	0.284	0.446
LSA	1-5B	300	1437	1159	12.9°	0.423	0.563
PPMI-SVD	2.5B	300	1869	1487	16.8°	0.331	0.466

Table 1: Results of the selected set of word embeddings.

The data needed to be split into the training and test set. The training set is used to find optimal transformation matrix $\mathbf{T}^*$ and optimal regularization weight α. The test set is used for the evaluation.

We manually selected the train set from the cities to evenly cover geographical area by the cities with the highest Wikipedia term frequency. The final train set contains 124 cities and the final test set contains 316 cities.

4.2 Word Embeddings

We chose a set of well-known word embeddings to show their differences using the proposed evaluation. In the following paragraphs we briefly introduce the chosen word embeddings.

SkipGram is a neural network based model (Mikolov et al., 2013b). Levy and Goldberg (2014) provide trained SkipGram models with two sizes of the context window (2, 5) and their own model that uses dependency-based context, denoted by SkipGram - BoW2, SkipGram - BoW5, and SkipGram - Dep, respectively.

GloVe is a log-bilinear model that tries to find word embeddings that are good at predicting global word co-occurence statistics (Pennington et al., 2014). We use embeddings provided by authors of the model trained on various corpus sizes (6, 42, and 840 billions words) and with various vector dimensions (50, 100, 200, 300).

FastText is an extension to SkipGram, where the word is represented as character n-grams (Bojanowski et al., 2017). We use embeddings provided by authors of the model trained on Wikipedia.

LexVec is based on factorization of positive point-wise mutual information matrix using proven strategies from GloVe, SkipGram, and methods based on singular value decomposition (Salle et al., 2016). We use two models provided by the authors of the model trained on Wikipedia and News Crawl (LexVec - w + nc), and Common Crawl (LexVec - cc).

MetaEmbeddings is an ensemble method that combines several embeddings (Yin and Schütze, 2016). We use the embeddings provided by the authors of the model.

WoRel is an extension of SkipGram, where a phrase (instead of a word) is used to guess the context words (Konkol, 2017). We use the model provided by the authors trained on Wikipedia and Gigaword corpus.

LSA is a count based method that creates a word-document co-occurrence matrix and reduces its dimension by singular value decomposition (SVD) (Landauer et al., 1998). We trained the models on Wikipedia.

PPMI-SVD creates word co-occurrence matrix where the co-ocurence is measured by positive

pointwise mutual information. The dimension of the matrix is then reduced by SVD. We used the hyperwords package (Levy et al., 2015) and trained it on Wikipedia and Gigaword corpus.

4.3 Evaluation

In our first experiment we evaluate the selected set of embeddings with the proposed evaluation metric. The results are shown in Table 1.

We provide results for two baselines. The first baseline (random placement) places cities randomly on the globe. The results for this baseline are computed analytically. The second baseline generates random embeddings, each value is selected randomly from uniform distribution between -1 and 1. The random embeddings are then evaluated in the same way as normal embeddings. The results show average results for five random embeddings. The comparison of the baselines show that the evaluation works as expected: random embeddings produce randomly placed cities.

The results show that all the evaluated word embeddings are significantly better than the baselines. This proves that the embeddings do not capture only the similarity between words but also nontrivial knowledge about the world.

Most geographic information was clearly captured by LSA, followed by FastText. A group of models, namely WoRel, SkipGram – BoW5, PPMI-SVD, and LexVec, achieved similar results and are only slightly worse than FastText. Surprisingly, GloVe (trained with similar amount of data) performed significantly worse. MetaEmbeddings achieved the worst results, probably because the ensemble was optimized for other purposes.

There is a high correlation between the performance in the mean geographic distance and Prec@10 measures. Models that are good at capturing global structure tend to be good at capturing local structure.

The type of the training data is probably more important than the size of the data. This can be seen on the LexVec models, where the model trained on Wikipedia and news articles outperforms the other model trained on significantly more data. Still, an extreme amount of data leads to good results as seen on the results of GloVe trained on various corpus sizes.

Recently, most of the NLP tasks use word embeddings based on local (window-based) context. Surprisingly, our evaluation shows that LSA, a method based on global (document-wide) context, outperforms all the other models in the proposed evaluation. The comparison of count based (PPMI-SVD) and predictive models (e.g. SkipGram, FastText) shows no significant differences between these two approaches.

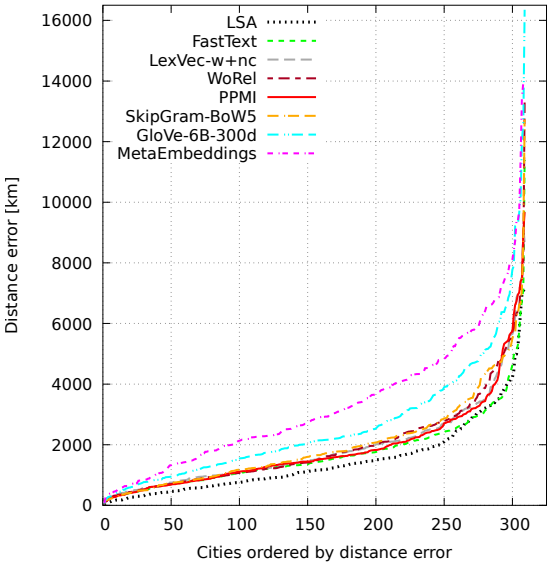

Figure 1: Distribution of distance errors.

Our evaluation shows that the mainstream models such as SkipGram and GloVe that perform similarly in intrinsic word similarity and extrinsic task based evaluations may have very different results in other types of evaluation.

4.4 Error Analysis

Figure 1 shows the distribution of geographic distance errors for individual cities. The distance error is reasonable ($\leq$ 2500 km) for approximately 90% of the cities for most of the word embeddings. Unfortunately, the rest of the cities has significantly larger error. In this section, we try to identify the source of the extreme errors.

Firstly, we suspected that the reason is sparseness and the extreme errors are caused by underrepresented words. In Figure 2 we show a relation between the number of occurrences of the city name in Wikipedia (training data for most of the methods) and the mean distance error. The word occurrences are equidistantly grouped into ten bins. We concluded that there is no clear relation between the number of occurrences of a city name and the distance error.

We also suspected ambiguity with common words. To check this hypothesis, we counted how

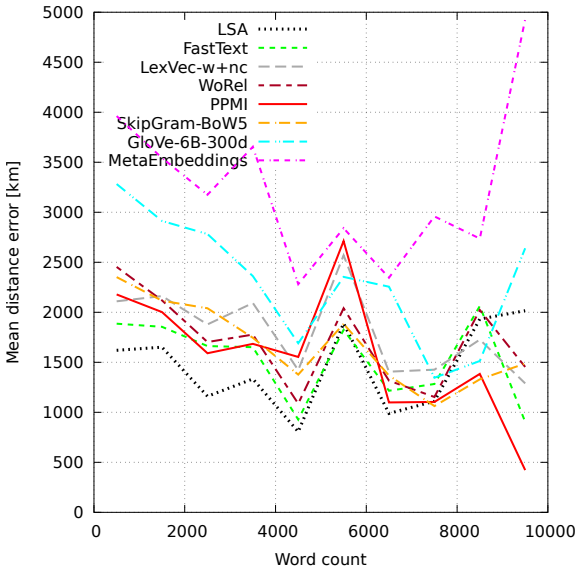

Figure 2: Relation between the number of occurrences of the city name and the distance error for LSA.

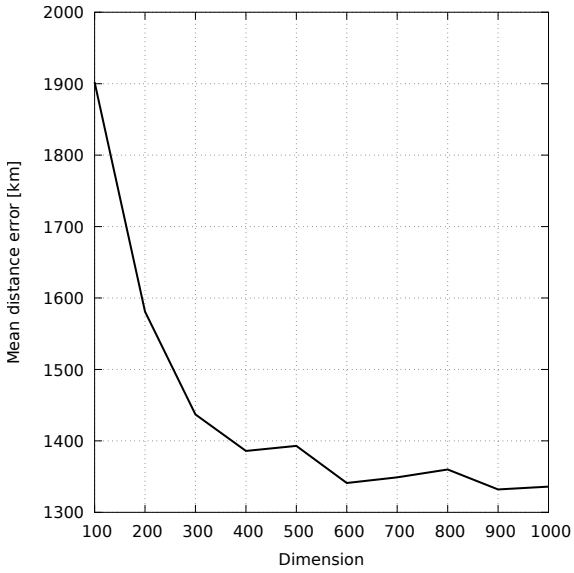

Figure 3: The relation between dimension of the vector space and the mean distance error for LSA.

many times the city name appears as lowercase and how many times with a capital letter. We found out that most of the words does not appear at all in lowercase version. A small portion of words has significant number of occurrences of the lowercase version (e.g. Phoenix), but they do not correlate with the distance error.

Lastly, we manually checked all the cities with extreme distance errors. We found out that the main problem is ambiguity with other named entities that are more famous than the city, e.g. the city Kobe is overshadowed by Kobe Bryant, Bismarck by Otto von Bismarck, Montgomery by the common first name. A special case of this problem is multiple cities with the same name. This is not a problem if there is large difference between the fame of the cities (e.g. London), but it is a problem for cities that are similar in size and fame (e.g. Midland, Kingstown, Bridgetown).

4.5 Embeddings Dimension

The dimension of the word embeddings obviously affects their results (Table 1). In this experiment we explore the effect of higher dimensions on the results. This should provide a hint to the authors of the semantic spaces how to choose the appropriate dimension.

In Figure 3, we show the results of LSA with dimension ranging from 100 to 1000. The performance degrades quickly as we decrease the dimension under 300. The results slightly improve as we increase the dimension from 300 to 600. There are no significant improvements as the dimension is increased over 600.

4.6 Regularization

The proposed evaluation uses regularization and requires the regularization weight α. Setting optimal regularization weight is difficult for some algorithms. We conducted an experiment to prove that the regularization weight does not play an important role in the evaluation, i.e. the scores of the embeddings are not heavily affected by our inability to find optimal regularization weights.

We performed randomized 10-fold cross-validation to find optimal regularization weight multiple times. The variance of the found regularization weights and also the impact of this variance were very small for a particular word embeddings method. Moreover, the optimal regularization weight is very similar for all the word embeddings. Figure 4 shows the mean geographic distance as a function of the regularization weight and suggests that the function can be easily optimized.

4.7 Noise Sensitivity

Given a set of models, the evaluation metric should be able to rank them reliably based on their quality. Batchkarov et al. (2016) propose a test of the reliability. They incrementally add noise to

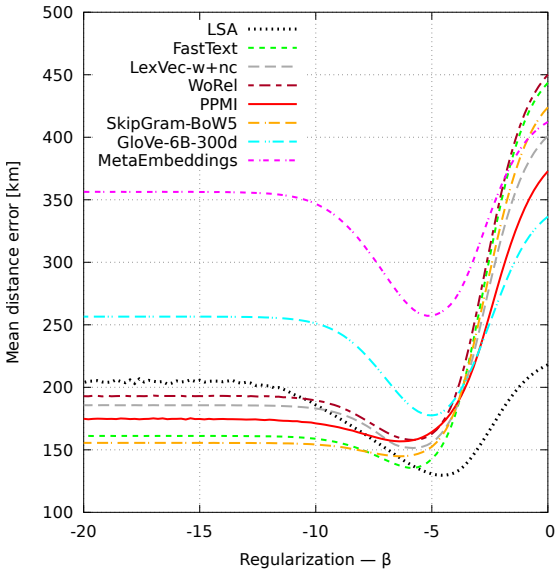

Figure 4: Influence of the regularization weight $\alpha = e^{\beta}$ on the mean geographic distance. The values are computed using 10 fold cross-validation on the training data.

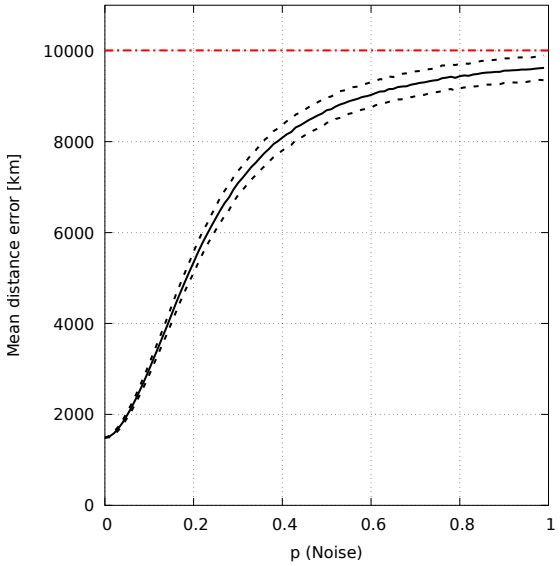

Figure 5: The effect of noise added to LSA embeddings. The figure shows mean value and standard deviation of 1000 runs. The red line on the top represents random city placement.

word embeddings and assume that word embeddings with more noise have lower quality. The metric should be able to capture the differences of the quality and smoothly and monotonically go from good results to results of random embeddings.

In Figure 5 we show the behavior of the proposed metric. We use the best embeddings (LSA) as a starting point. Then we add noise uniformly sampled from interval $[0, p]$ to each value in the embeddings. The parameter p is incrementally increased with step 0.01 from 0 to 1. For each value of p we repeat the evaluation 1000 times. The proposed metric works as expected. Firstly, the mean distance error almost linearly increases. As the embeddings become more random the increases slow down until they converge to the results of random embeddings.

4.8 Visualization

As a side effect, our evaluation approach also produces a natural visualization presented in Figure 6. The visualization can be used for comparison of methods, error analysis, or demonstration of semantics and unsupervised learning. The transformation also allows us to visualize common words on the map, not only city names.

5 Conclusion

We have proposed a new evaluation method for word embeddings. It measures how much information about geographic location of cities is contained in word embeddings. This type of evaluation differs from previously presented evaluations and forms a new word embeddings evaluation paradigm. The new paradigm does not evaluate the embeddings from the natural language processing view, but rather from the artificial intelligence view, where the algorithm tries to capture some information about the world.

We have analyzed both the evaluation metric and commonly used embeddings. We have shown that the metric is stable and can reliably distinguish between good and poor models.

LSA achieved the best results with mean geographic distance error of 1437 kilometers. Surprisingly, it outperformed mainstream models such as SkipGram. GloVe, with state-of-the-art results from other evaluations, performed rather poorly in the proposed evaluation.

In the future, we would like to implement the proposed paradigm with other similar evaluations, where we try to find out if the model is able to capture a specific real-world information.

The dataset and the evaluation software can be

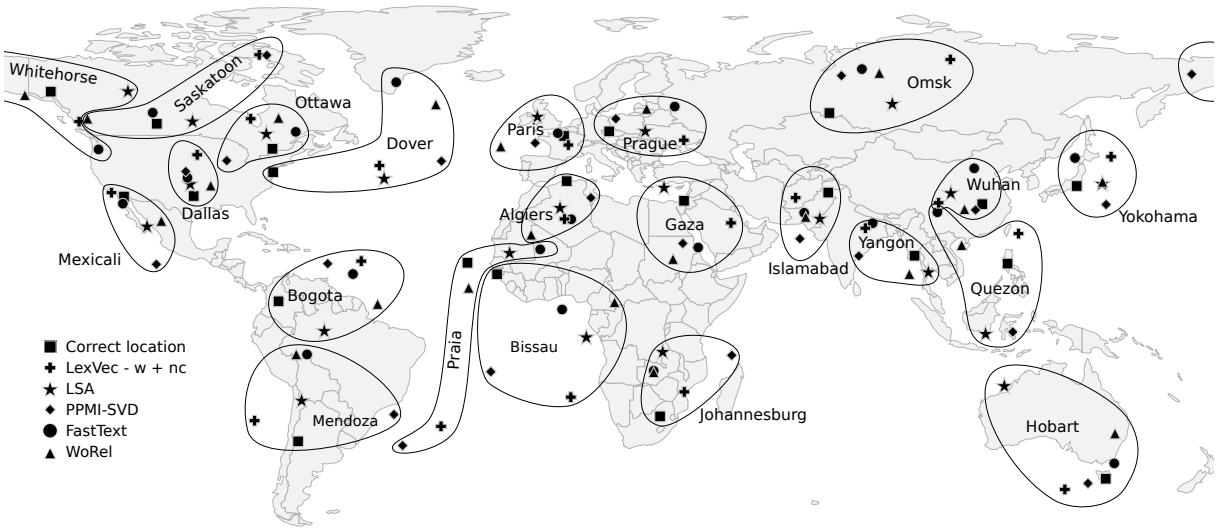

Figure 6: A visualization of selected cities placed on the map based on various word embeddings. Each circle denotes one city.

downloaded from the authors' websites[1].

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I and by the university specific research project SGS-2016-018 Data and Software Engineering for Advanced Applications.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815, Baltimore, Maryland. Association for Computational Linguistics.

[1]Currently at `konkol.me` and `nlp.kiv.zcu.cz`

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. 2016. A critique of word similarity as a method for evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 7–12, Berlin, Germany. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *The First Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. 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*, WWW '01, pages 406–414, New York, NY, USA. ACM.

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.

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

Michal Konkol. 2017. Joint Unsupervised Learning of Semantic Representation of Words and Roles in Dependency Trees. In *RANLP 2017 – Recent Advances in Natural Language Processing*.

Michal Konkol, Tomáš Brychcín, and Miloslav Konopík. 2015. Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470 – 3479.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany. Association for Computational Linguistics.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 104–113.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

Neha Nayak, Gabor Angeli, and Christopher D Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. In *The First Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Alexandre Salle, Aline Villavicencio, and Marco Idiart. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–424, Berlin, Germany. Association for Computational Linguistics.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398. ACL.