Evaluation of Vector Transformations for Russian Static and Contextualized Embeddings

Olga Korogodina [1], Vladimir Koulichenko [1], Olesya Karpik [2] and Eduard Klyshinsky [1]

[1] *National Research University Higher School of Economics, Myasnitskaya 20, Moscow, 101000, Russia*
[2] *Keldysh Institute of Applied Mathematics, Miusskaya sq., 4, Moscow, 125047, Russia*

Abstract

The authors of Word2Vec claimed that their technology could solve the word analogy problem using the vector transformation in the introduced vector space. By default, the same is true for both static and contextualized models. However, the practice demonstrates that sometimes such an approach fails. In this paper, we investigate several static and contextualized models trained for the Russian language and find out the reasons of such inconsistency. We found out that words of different categories demonstrated different behavior in the semantic space. Contextualized models tend to find phonological and lexical analogies, while static models are better in finding relations among geographical proper names. In most cases, the average accuracy for contextualized models is better than for static ones. Our experiments have demonstrated that in some cases the length of the vectors could differ more than twice, while for some categories most of the vectors could be perpendicular to the vector connecting average beginning and ending points.

Keywords

Word Embeddings, Vector Space, Vector Transformation, Word Analogies

1. Introduction

Vector models, originally introduced in paper [1] in 2003, boost progress in NLP area. The main idea of embeddings is generation of fixed-size vectors according to statisti-cal information about the word context by means of a neural network. This concept was developed in [2] where the authors demonstrated that such pre-trained vectors could be useful for solution of different natural language processing problems. The real revolution was made by the Word2Vec model, introduced in 2013 [3 5], which is based on the distributive hypothesis. The Word2Vec model also uses neural networks reinforced by several new ideas.

First of all, the new approach [4] had less computational complexity compared to the previous systems. The next article [5] increases its learning rate and accuracy. The greatest contribution of the authors was the publication of source codes and pre-trained language models for free use.

The FastText model, which was introduced in 2017 [6], improves some drawbacks of Word2Vec using the following ideas. Both prefixes and postfixes of words carry semantic information as well as word roots. In this case, the meaning of a word can be composed of the meaning of its parts. Dividing a word into character n-grams, the system collects more information about the same n-gram using contexts of different words. Thus, the FastText model doesn't need lemmatization and can use relatively smaller corpora to achieve the same outcome.

Both the Word2Vec and FastText models have a big drawback: they use all words from a context of a considered word. However, the considered word can have several meanings; often, every meaning of a word has its own set of contexts which are slightly intersecting or have no intersection at all. Thus, training a model, one should try to separate those meanings into different vectors.

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia
EMAIL: eklyshinsky@hse.ru (E. Klyshinsky); parlak@mail.ru (O. Karpik)
ORCID: 0000-0003-3601-4677 (O. Korogodina); 0000-0003-3256-8955 (V. Koulichenko); 0000-0002-0477-1502 (O. Karpik); 0000-0002-4020-488X (E. Klyshinsky)

Such drawbacks were corrected by the ELMo (Embeddings from Language Mod-els) [7] and BERT (Bidirectional Encoder Representations from Transformers) [8] methods, presented in 2018. Both of these models use Transformer neural networks with several attention layers, but in distinct of ELMo, BERT uses bidirectional layers. Unlike static models such as Word2Vec and FastText, which always return the same vector, the contextualized models, BERT and ELMo, return a vector according to the meaning of the word in the given context. It makes some problems, since a fuzzy comparison of vectors rather than checking their equality should be conducted; how-ever, in most cases contextualized vectors lead to higher productivity. The paper [9] provides a good overview and description of vectors contextualization.

The authors of [3-5] argued that the new semantic space allows vector arithmetic. Their example of "Queen = King – man + woman" swiftly becomes very famous. However, it becomes clear in a short time that such operations do not always lead us to success. One of the proofs of this concept is the problem of words analogies. The early experiments demonstrated that another favorite example, countries and their capitals, did not work correctly for any example. The accuracy of this analogy was quite high but not enough to state that vector arithmetic worked properly. However, the problem of words analogies operates not as good as it could. Despite the fact that embeddings allow correct finding of a list of semantic neighbors for a given word, what makes it a crucial part of modern systems of natural language processing, the authors of [10] demonstrated that results could dramati-cally vary depending on used task and model.

As it was demonstrated in the paper [11], the quality of the word analogies prob-lem depends on the considered category and pre-trained static model. The authors of [12] investigate the word analogies problem as a task of reflection. They demonstrat-ed that for the same analogy there could be several mirrors responsible for their own region of the considered semantic space. The reason is that different word groups can have different meanings for the same analogy, i.e. transition vectors for these groups will be different as well. For example, gender differences in a regular and royal family have different connotations, which differ from professional gender variations.

The aim of this paper is to conduct experiments from [11] for contextualized BERT and ELMo models, compare and generalize the results for static and contex-tualized models, and find out whether contextualized models provide any ad-vantages in comparison with static models.

The rest of the paper is organized as follows. In Section 2, we state the problem of word analogies as a vector transformation problem. Sections 3 and 4 describe the used data set and the numerical evaluation of language models for the Russian lan-guage. Section 5 analyses the results achieved and compares the results for static and contextualized models. Section 6 concludes the article.

2. Formal Statement of the Problem of Word Analogies

In general terms, the main question of the problem of word analogies could be stated as "Is there a word *c* which relates to the word *b* as the word *a*' relates to the word *a*?" To answer this question, semantic embeddings use a vector representation of these words. Let $v_{a'}$ and v_a be vectors corresponding to the words *a*' and *a* respectively; in this case, the vector difference $v_{a'} - v_a$ expresses the semantic relation (or in other words, the semantic difference) between words *a* and *a*'. Thus, in order to find an analogue, we should find the word *x* and its corresponding vector *y* in such a way that $y - v_b = v_{a'} - v_a$, or

$$y = v_b + v_{a'} - v_a. \quad (1)$$

However, the probability of existence of a word having exactly the same vector as v_x is extremely small. That is why we will find vector *y*' that is the closest word to the vector *y*:

$$y' = \underset{v \notin \{v_a, v_{a'}, v_b\}}{argmax}\ cos(v, v_b + v_{a'} - v_a) \quad (2)$$

We can reformulate the question for word groups. Let us consider a set of word pairs (w_{11}:w_{12}), (w_{21}:w_{22}), …, (w_{N1}:w_{N2}) that have the same semantic or lexical relation, and their corresponding vectors v_{11}, v_{11}, v_{21}, v_{21}, …, v_{N1}, v_{N1}. In this case, the task of word analogies could be formulated as following: if there is a vector *x* that makes an affine transformation of w_{11}, w_{21}, …, w_{N1} to w_{12}, w_{22}, …, w_{N2}, then *x* is such that

$$v_{i2} = \underset{v'}{argmax} cos(v', v_{i1} + x). \quad (3)$$

Let us denote the fact that the word *a* relates to the word *b* in the same sense as the word *c* relates to the word *d* by the following equation: (*a*:*b*) :: (*c*:*d*). For example, (*apple*:*fruit*) :: (*cucumber*:*vegetable*), (*apple*:*apples*) :: (*cucumber*:*cucumbers*), and, classical, (*king*:*queen*) :: (*man*:*woman*). In this case, a request to find an analogy looks like (*king*:*?*) :: (*man*:*woman*) or (*man*:*woman*) :: (*king*:*?*).

The task of word analogies is very sensitive to the noise in the input data. A word can be homonymous; this means that it should be presented as two or more separate vectors reflecting different meanings of this word. In case of Word2Vec, such a word will be represented only by a vector that will be a superposition of all its meanings. Moreover, the resulting vectors of similar entities could express differences in their occurrence with other words. For example, a dog and a cow are both animals, but dog is a carnivore and a human's friend, while a cow is an herbivore and gives milk; thus, the analogy is not complete here. Contextualized models improve static models using a context to infer the real sense, and a word resulting vector. However, the vectors for the same word in slightly different contexts will not be equal. In order to eliminate such influence, the authors of the 3CosAvg method [13] introduce a new formula that takes into account not just a pair of words but two whole groups having the same analogy:

$$y' = \underset{v \in V \setminus \{v_b\}}{argmax} cos\left(v, v_b + \frac{\sum_{i=1}^{n} v_{a'_i}}{n} - \frac{\sum_{i=1}^{n} v_{a_i}}{n}\right). \quad (4)$$

As it was shown in [14, 15], methods Only-b, Ignore-a, and PairDirection give unsatisfactory results. Unlike [10], we will use the 3CosAvg method only, since [11] has demonstrated that this method provided more robust results which were less dependent on biases in input vectors and their polysemy. We have not tested the X2Static method [16], which calculates the average vector for all embeddings returned by a contextualized model, because of its novelty. However, we believe that it should not dramatically increase performance of the words analogies method, since the 3CosAvg method averages such vectors in the same way.

3. Used Data Sets

For static embeddings, we used several pre-trained models from the site RusVectōrēs (http://rusvectores.org/ru/models/): Araneum Upos Skipgram 2018, Ruwikiruscorpora Upos Skipgram 2018, Ruwikiruscorpora Upos Skipgram 2019, Tayga Upos Skipgram 2019, News Upos Skipgram 2019, Ruscorpora Upos CBOW 2019, Araneum Fasttext Skipgram 2018, Facebook FastText CBOW 2018 [17, 18]. The first models were trained using Word2Vec, the two later ones were trained using FastText. For contextualized embeddings, we used RuBERT 2019 [19], Sentence RuBERT 2019, Conversational RuBERT 2019, ELMo Ru Wiki 2019, ELMo Ru News 2019, and ELMo Ru Tw 2019. These models were selected as they tagged a text by words but not by parts of words, as some newer models did.

Note that contextualized models need a context; thus, there is no possibility for passing merely a word for acquiring a resulting vector. That is why we used a collection of news wire texts. Every contextualized vector was calculated as a mean value of vectors, calculated by manually selected texts. The set of selected examples could influence resulting vector but could not lead to misrepresentation of the whole picture.

For semantic analogies, we used the Russian versions of Google analogy test set [4] and BATS (The Bigger Analogy Test Set) [13]. These data sets was translated and extended by a human expert. For grammatical analogies we used morphological dictionary of the Russian language. The list of used categories presented in Table 1.

Table 1
Used semantic and grammatical categories

Category	ID	Example		Number of Pairs
Famous capital → Country	A1	Афины	Греция	23
All capitals → Country	A2	Канберра	Австралия	115
Country → currency	A3	Ангола	кванза	30
Country → Adjective	A4	Австралия	австралийский	41
Country → Language	A5	Аргентина	испанский	36
Masculine → Feminine	A6	наследник	наследница	67
Singular → Plural	A7	улыбка	улыбки	100
Antonyms with *не-*(*non-, ir-*)	A8	определенный	неопределенный	27
Adjective → Adverb	A9	спокойный	спокойно	30
Possessive Adjective → Comparative Adjective	A10	яркий	ярче	24
Verb → Corresponding Noun with *-ация* (*-ation*)	A11	консультировать	консультация	55
Verb → Corresponding Noun with *-ение* (*-ment, -ion*)	A12	назначать	назначение	55
Verb → Corresponding Noun with *-тель* (*-er, -or*)	A13	слушать	слушатель	56
Verb → Reflexive Verb	A14	откопаю	откопаюсь	400
Verb → Verb with *при-*	A15	вязать	привязать	376

4. Evaluation

For the evaluation purpose, we calculated the accuracy metrics for all categories as well as for language models. Fig. 1 and 2 demonstrate results for the static and contextualized models, respectively, calculated by the 3CosAvg method. Dark blue shows results with a higher accuracy, up to 1; light blue shows results close to zero. The results from Fig.1 were taken from [11]. Note that for contextualized models we have not calculated the last two categories since they demonstrated low accuracy for static models.

Obviously, contextualized models demonstrate better accuracy than static ones. For static models, there are only five model-category combinations which accuracy exceeds 0.9, while for contextualized ones there are about 40% of such combinations which overpass this threshold. There are only three categories where static models hit contextualized: Capital → Country for famous (A1) and all (A2) countries, and Country → Language (A5). Note, that both types of models show low accuracy for the A5 category. The same is true for the category A3 Country → Currency.

We found that RuBERT Sentence exceeds other BERT models for each category in our task. The productivity of ELMo models depends on a category. In case of grammatical parameters, ELMo Ru Twitter overpasses other models, while for information about countries ELMo Ru Wiki wins in most cases.

We can state the hypothesis that the analyzed words of such categories as Capital → Country, Country → Language, and Country → Currency have several meanings. For example, the name of the capital of a well-known country can be used as the name of the proper city (*An accident in Moscow*) and as a synonym of the corresponding country (*Moscow plays muscles again*). There are also several countries which share the same language or currency. Such a variety makes analysis more difficult for contextualized models which have fewer contexts for each separated meaning. It's easy to see that models trained on news wire or Wikipedia texts demonstrate better solutions for categories entailed to countries. Though RuBERT was trained on Wiki and news texts, it demonstrates worse productivity for country categories than other BERT models. However, other models are fine-tuned versions of the RuBERT model; thus, they had extra contexts to learn.

Figure 1: Accuracy by 3CosAvg method for static models [11]

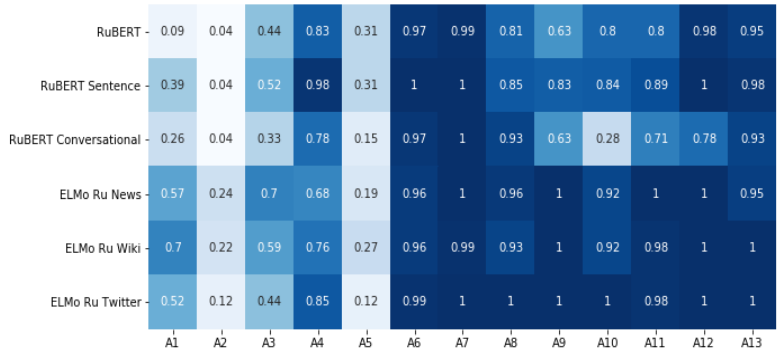

Figure 2: Accuracy by 3CosAvg method for contextualized models

5. Data Analysis

In order to find out the reasons of success and fail, we conducted a visual analysis of the resulting vectors. First of all, we projected 300-dimensional vectors into 2D-space using Principal Component Analysis (PCA). Instead of t-SNE and UMAP, PCA does not create areas with non-linear skew. As a result, parallel vectors keep their parallelism. On the other hand, there is a chance that two arbitrary vectors placed on parallel planes could become parallel on the projection; however, such distortions in the data are not very critical.

For our experiments, we used two static language models which were trained on the same Araneum corpus: Araneum Upos Skipgram 2018 and Araneum Fasttext Skipgram 2018. For contextualized models, we used Ru BERT and ELMo Ru News which are not the best solution for the selected category A1 Famous capital → Country. Fig. 3, 4 present five randomly selected word pairs for these four models.

The main reason of low accuracy in the task of word analogies is the bias between the vectors. It is easy to see that the Word2Vec vectors on Fig. 3 are mostly parallel, excluding slightly bias for *Оттава-Канада* (*Ottawa-Canada*). The length of the vectors is also almost the same. Averaging among the start and end points of the vectors helps to adjust these small biases in the Word2Vec model. However, the FastText model is oriented mostly on word parts; that is why its vectors are almost randomly oriented and have no preferential direction. Contextualized models show almost the same picture (Fig. 4): the vectors are mostly parallel for ELMo model, but for the RuBert model this is not true.

Since PCA makes some distortions and allows us to analyze a projection instead of raw data, we decided to draw the distribution of angles among vectors. Fig. 5, 6 demonstrate cosine distances for vectors belonging to FastText and ELMo Ru Twitter models. We selected the most representative categories which show two different situations; however, in both cases the results were unsatisfactory. On the left picture, all vectors are non-parallel, the cosine of angles among the vectors is less than 0.75. A worse situation is drawn on the right figure: some vectors have opposite directions and their cosine reaches -0.75.

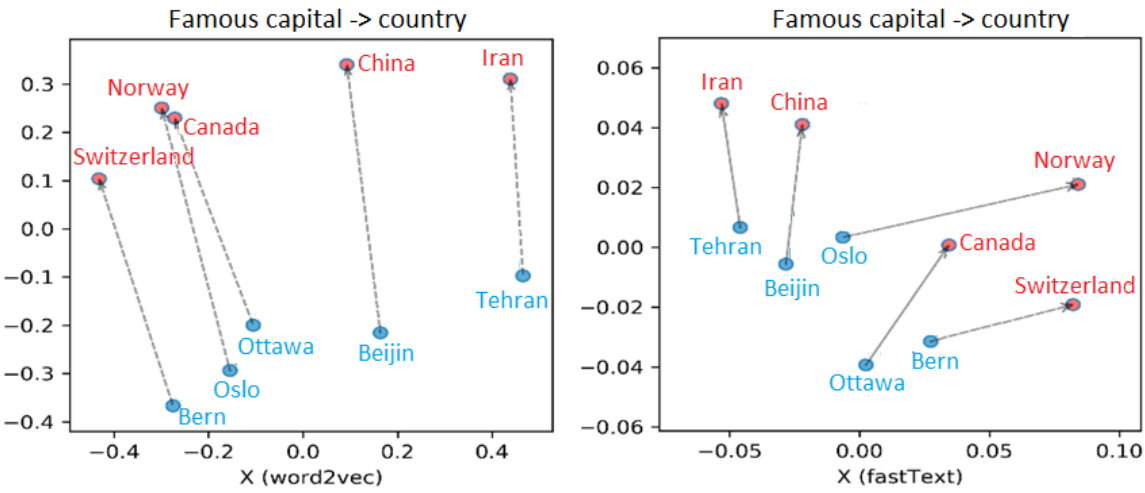

Figure 3: Word2Vec and FastText vectors for the category A1 Famous capital → Country

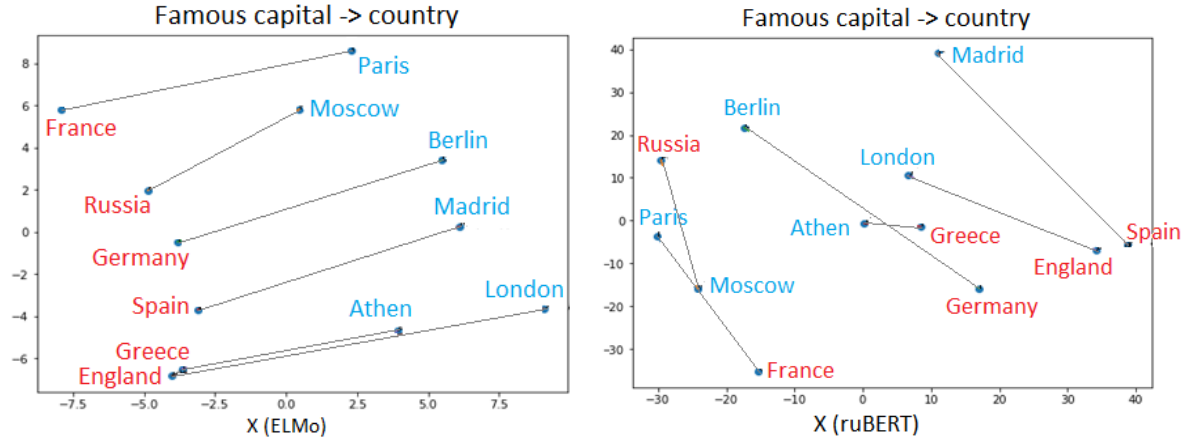

Figure 4: ELMo Ru News and RuBert vectors for the category A1 Famous capital → Country

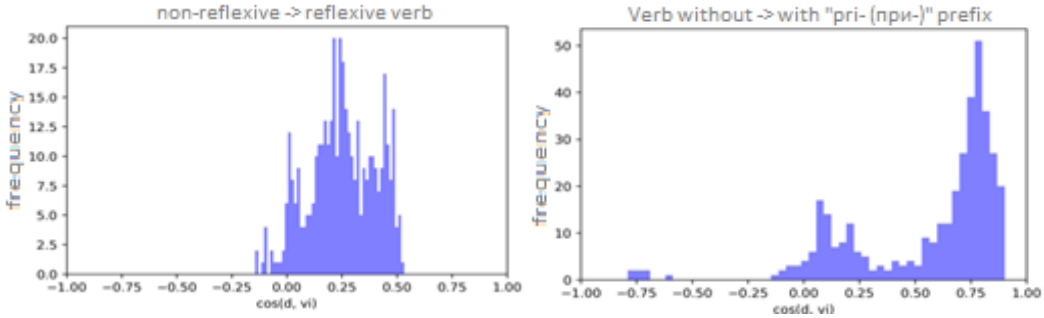

Figure 5: Histogram for cosine similarity among average vector and vectors for the categories A14 and A15, Araneum FastText model

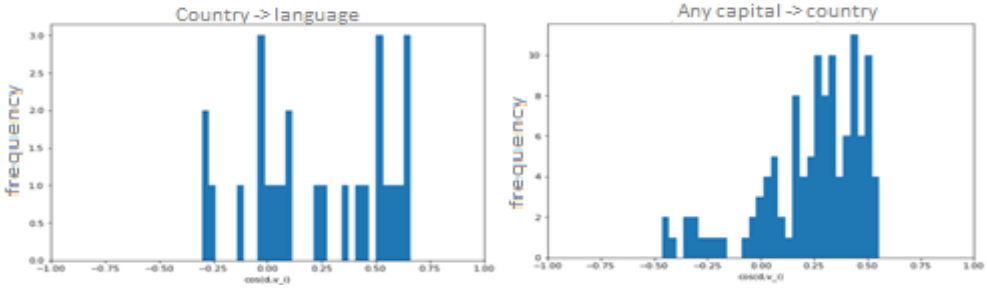

Figure 6: Histogram for cosine similarity among average vector and vectors for the categories A5 and A2, ELMo Ru Twitter model

Another reason for such drawbacks in the results is the length of the vectors. Even if the vectors' directions are the same, but their lengths significantly differ, then the words analogies task will fail. Thus, we visualized both the vector length and its direction according to the average transition vector; the start and end points of transition vector were calculated as average among the start and end points of the original vectors (Fig 7). We present here only a few of the most representative figures for the Araneum FastText static model which demonstrate the reasons of failures. Pink dots represent correctly resolved analogies, the blue dots represent the opposite situation.

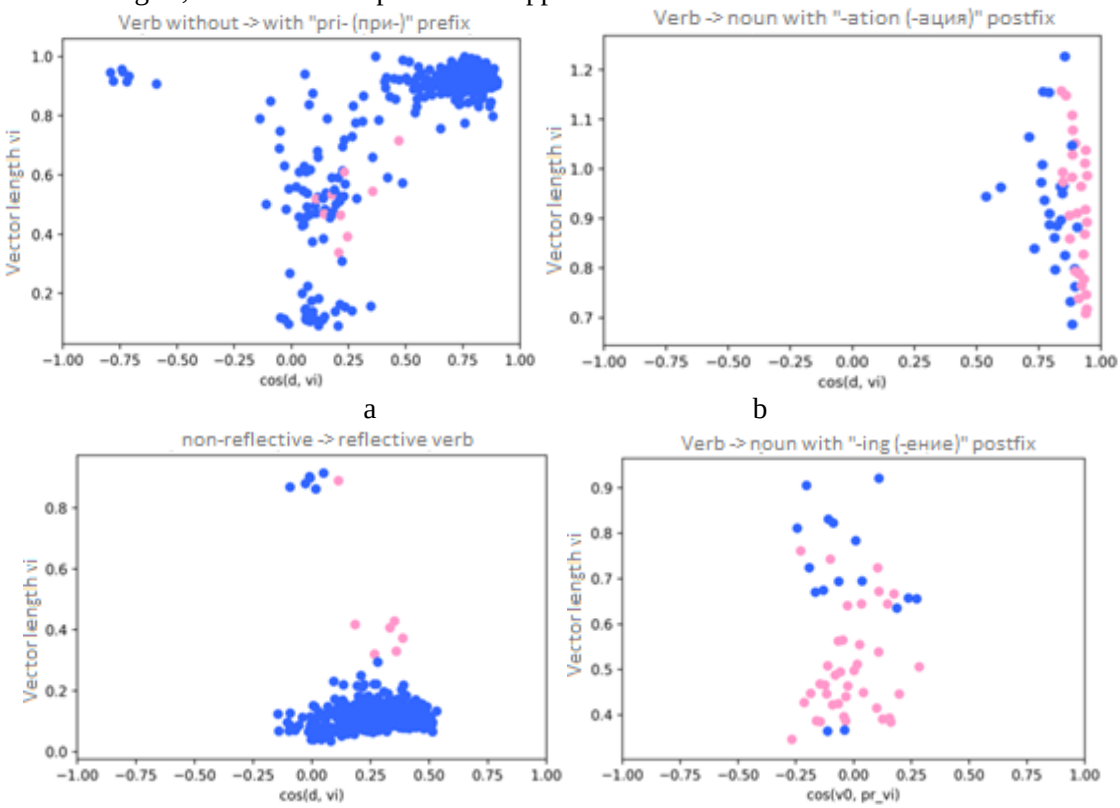

Figure 7: Relations between vector length and its angle with an average vector

Fig. 7a represents the case when half of the vectors are parallel and have the same length, and half of the vectors have a different direction, which is perpendicular to the average vector in most cases, and different length. This situation corresponds to 0.03 accuracy since there are several vectors parallel to the average vector. Figure 7b represents the situation when all the vectors are mostly parallel but have different length (accuracy = 0.53). Fig. 7c corresponds to the situation when the vectors have almost the same size but they are perpendicular (accuracy = 0.02). Finally, fig. 7d represents perpendicular vectors of different length (accuracy = 0.71).

We examined the fluctuations found on fig. 6 and 7 but did not find any regularities in the word sets. Probably, the resulting noise can be explained by the fact that there are several semantic clusters joined into one.

Our results correspond to the idea of parameterized reflections described in [12], but we did not investigate this version. However, Fig. 6 and 7 demonstrate some clusters which can be considered as candidates to application of such reflection planes. Anyway, the main idea of [12], that there are several dedicated directions for some categories, seems reasonable in the light of our experiments. Moreover, it is true not only for static but for contextualized models as well, but not for all of them, since different models reflect different spatial situations.

6. Discussion and Conclusion

In this paper, we found several reasons why the vector transformation does not work on some categories of word analogies.

1. The used language model should be taught on texts that have enough occurrences of words for which the task of word analogies is solved. That is true for both static and contextualized models, but static models more affect on the selected text corpora. On the other hand, contextualized models need larger corpora for better accuracy in more polysemous categories. As we can see in Fig. 1 and 2, categories that include countries, their capitals, and other related words are better analyzed using corpora such as Wikipedia and news wire, since such corpora contain enough information for inference of logical relations among these words.
2. Following [10] we can state that quality of results depends on the model in hand. Fig. 1 and 2 demonstrate that the result depends on the corpus used for the model training and on the selected domain. However, contextualized models have a great preference; they can be fine-tuned on the analyzed corpus. As a result, the quality of the solution should increase.
3. In a common case, the task of word analogies could not be solved using a random pair of words taken from two investigated categories. For example, Moscow and Berlin are respectful representatives of their countries in case of international or cultural affairs; these cities are used as synonyms of Russian and German government and culture. However, Bogota and Kampala are the capital rather than the government. Thus, there will be several preferential directions for different prefix or suffix values. If someone misuses just one word in a pair, he or she will get the wrong transition vector. Averaging of the start and end points helps to eliminate this problem, but this method has some drawbacks. One should have a list of words in a given category to average their vectors; this is not always possible. On the other hand, sometimes such a list is available, and one could use it to tune vectors in order to predict out-of-list words. But previously he or she should be sure that this list does not contain several preferred semantic directions.
4. The main idea of an affine transition is that there is one vector that could be added to the word *a* to find its analogy *b*. It means that all vectors for *a*:*b* should be equal, i.e. have approximately equal length and orientation angles. At least, the value of the bias between this transition vector and the correct answer vector should be less than half the distance to the nearest neighbor. As we found out, this is not always true. In the case of homonymous prefixes and suffixes, some vector groups could be oppositely directed. This means that the word analogies task could not be solved using only one transition vector. Our experiments have demonstrated that in some cases the length of the vectors could differ more than twice. Such biases lead to the situation when the software module must generate several outputs, and the user has to find some extra methods to find the correct answer.
5. Contextualized models provide more robust solution in case of rich mono-thematic (or general purpose) corpora. Note, that some specific tasks have better solution with static models, but the best default solution is to use contextualized models.

Our method of analysis of affine transformations for embedding vectors could be used as a method of exploratory analysis of domain. Before using of method of vector analogies, a researcher should check if such analogies could be successfully applied to a selected domain or with a selected language model, both pre-trained and trained on domain texts. This will help eliminate some mistakes and evaluate further results in advance.

7. References

[1] Y. Bengio, R. Ducharme, P. Vincent, P.A. Jauvin, Neural Probabilistic Language Model, Journal of Machine Learning Research 3 (2003) 1137–1155.

[2] R. Collobert, J. Weston, A unified architecture for natural language processing, in: Proceedings of the 25th International Conference on Machine Learning, 2008, vol. 20, pp. 160–167.

[3] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic Regularities in Continuous Space Word Representations. In: Proc. of HLT-NAACL, 2013, pp. 746-751.

[4] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proc. of International Conference on Learning Representations (ICLR), 2013.

[5] T. Mikolov, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: Proc. of 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 3111-3119.
[6] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, Transactions of the Association for Computational Linguistics 5 (2017) 135-146.
[7] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, ArXiv:1802.05365, URL: https://arxiv.org/pdf/1802.05365.pdf.
[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv:1810.04805, https://arxiv.org/pdf/1810.04805.pdf.
[9] K. Ethayarajh, How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings, ArXiv:1909.00512v1, https://arxiv.org/pdf/1909.00512.pdf.
[10] B. Wang, A. Wang, F. Chen, Y. Wang, J. Kou, Evaluating word embedding models: methods and experimental results, in: APSIPA Transactions on Signal and Information Processing, 2019, 8. doi: 10.1017/ATSIP.2019.12
[11] O. Korogodina, O. Karpik, E. Klyshinsky, Evaluation of Vector Transformations for Russian Word2Vec and FastText Embeddings, in: Proc. of Graphicon 2020. doi: 10.51130/graphicon-2020-2-3-18
[12] Y. Ishibashi, K. Sudoh, K. Yoshino, S. Nakamura, Reflection-based Word Attribute Transfer, ArXiv:2007.02598v2, URL: https://arxiv.org/pdf/2007.02598.pdf.
[13] A. Drozd, A. Gladkova, S. Matsuoka, Word Embeddings, Analogies, and Machine Learning: Beyond King-Man+Woman=Queen, in: Proc. of COLING 2016, pp. 3519–3530.
[14] O. Levy, Y. Goldberg, Linguistic Regularities in Sparse and Explicit Word Representations, in: Proc. of 18th Conf. on Computational Natural Language Learning, 2014, pp. 171-180. doi: 10.3115/v1/W14-1618
[15] T. Linzen, Issues in evaluating semantic spaces using word analogies, in: Proc. of 1st Workshop on Evaluating Vector-Space Representations for NLP, 2016, pp. 13–18.
[16] P. Gupta, M. Jaggi, Obtaining Better Static Word Embeddings Using Contextual Embedding Models. arXiv:2106.04302v1, URL: https://arxiv.org/pdf/2106.04302.pdf.
[17] A. Kutuzov, E. Kuzmenko, WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models, in: Analysis of Images, Social Networks and Texts (AIST), 2016, vol. 661, pp. 155-161. doi: 10.1007/978-3-319-52920-2_15
[18] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning Word Vectors for 157 Languages, in: Proc of LREC'2018, 2018, pp. 3483-3487.
[19] Y. Kuratov, V. Arkhipov, Adaptation of Deep Bidirectional MultilingualTransformers for Russian Language. arXiv:1905.07213v1, URL: https://arxiv.org/pdf/1905.07213.pdf.