Costra 1.1: An Inquiry into Geometric Properties of Sentence Spaces*

Petra Barančíková[0000−0002−4070−2766] and Ondřej Bojar[0000−0002−0606−0050]

Charles University, MFF ÚFAL
{barancikova,bojar}@ufal.mff.cuni.cz

Abstract. In this paper, we present a new dataset for testing geometric properties of sentence embeddings spaces. In particular, we concentrate on examining how well sentence embeddings capture complex phenomena such paraphrases, tense or generalization. The dataset is a direct expansion of Costra 1.0 [7], which we extended with more sentences and sentence comparisons. We show that available off-the-shelf embeddings do not possess essential attributes such as having synonymous sentences embedded closer to each other than sentences with a significantly different meaning. On the other hand, some embeddings appear to capture the linear order of sentence aspects such as style (formality and simplicity of the language) or time (past to future).

Keywords: sentence embeddings · sentence transformations · paraphrasing · semantic relations

1 Introduction

Trained vector representations of words and sentences, known as embeddings, have become ubiquitous throughout natural language processing (NLP). Since their popularity took off with the introduction of word2vec word embeddings [15], numerous different methods with different properties have emerged, highlighting the importance of estimating their quality. However, it is not entirely clear in which way the embeddings should be evaluated, aside from the performance in the task they originate in. Two main classes, *extrinsic evaluation* and *intrinsic evaluation*, are considered [17].

Extrinsic evaluation utilizes word embeddings as feature vectors for machine learning algorithms in downstream NLP tasks. It serves well in choosing the best method for a particular task but not as an absolute metric of embedding quality as the performances of embeddings do not correlate across different tasks [5].

[16] demonstrate the presence of linguistic regularities in the word2vec embedding space. Namely, they show that various word analogy tasks can be solved

* This research was supported by the grants 19-26934X (NEUREM3) of the Czech Science Foundation, 825303 (Bergamot) of the EU, and SVV project 260 575. This work has been using language resources stored and distributed by the project No. LM2015071, *LINDAT-CLARIN*, of the Ministry of Education, Youth and Sports of the Czech Republic.

by simple vector arithmetic in the embedding space, e.g. finding correct word D for words A, B, C and their respective embeddings v_A, v_B, v_C by optimizing:

$$\underset{D \in V \setminus \{A,B,C\}}{\arg\max} \quad \mathrm{sim}_C(v_D, v_A - v_B + v_C), \tag{1}$$

where sim_C represent cosine similarity between two vectors. This works for various semantic and syntactic relationships, like for example:

A	B	C	D
king	man	woman	queen
Russia	Moscow	Paris	France
walked	walk	tell	told
bigger	big	small	smaller

This lead to a novel approach—intrinsic evaluation—in which word embeddings are compared with human judgment on word relations. There is a large number of available datasets for syntactic and semantic intrinsic evaluation, word analogy task [15, 16] belongs among the most popular methods.

For sentence embeddings this is a different story. When Kiros et al. [12] introduced Skip-Thought vectors, they evaluated their quality in eight supervised tasks such as paraphrase detection or sentiment polarity. This extrinsic evaluation or transfer tasks became the de facto standard for evaluation and comparison of sentence embeddings, despite the fact that even simple bag-of-words (BOW) approaches often achieve competitive results on transfer tasks [18].

[1, 11] introduce intrinsic evaluation of sentence embeddings, however, most of the research in interpretation of sentence embeddings consists of probing for surface linguistic features of the sentence such as its length, verb tense, word order, etc. Furthermore, [14, 4] indicate that strong performance in these tasks might be caused by test flaws—the test sentences are grammatically too simple.

However, any geometric properties of an embedding space remain a largely uncharted territory. We attempt to fill this gap, examining whether sentence representation spaces exhibit regularities with regard to certain kinds of relationships, in a way similar to the linear relations observed in word vector spaces.

To this end, we devise a new dataset on the basis of Costra 1.0 [7], which we extend with information on linear ordering of embedded sentences with regard to certain kinds of relationships. These allow us to test empirically whether existing sentence embedding models reflect analogical relationships between sentences.

The paper is structured as follows: Section 2 presents existing methods of semantic evaluation of sentence embeddings and available off-the-shelf embeddings. Section 3 describes the methodology for constructing our dataset. Section 4 details the evaluation of embeddings and Section 5 presents the results.

2 Related Work

2.1 Sentence Embedding Space Evaluation

Zhu et al. [21] compare sentence embeddings from a relational perspective using automatically generated triplets of sentence variations and explore how syntactic or semantics changes of a given sentence affect the similarities among their sentence embeddings. The following example sentences illustrate this point:

S1: A pig is eating goulash.

S2: A pig is feeding on goulash.

S3: A pig is not eating goulash.

Synonyms (S1,S2) should be embedded closer to each other in a vector space than sentences with similar wording but different meaning (S1,S3) and (S2,S3). They discover that several embeddings perform surprisingly well in these tasks.

A sentence analogy task was recently introduced in [21]: in template sentences they substitute a pair of words such as state/capital, man/woman or plural/singular. To test, whether the embeddings are really able to find the analogy correctly, they create incorrect sentences similar in wording to the correct ones and examine whether Equation (1) finds the correct sentence.

Similarly, [6] examined sentences that are close in wording but differ in one key aspect (e.g. change of gender, adding an adjective, removing a numeral) and show that the changes form meaningful clusters in the sentence vector space.

In Costra 1.0 [7], we attempted to move to more sophisticated types of sentence relations, beyond those in [20, 6]. We present a dataset of complex sentence transformations in Czech. It is created manually with the aim to thoroughly test how well sentence embeddings capture the meaning and style of sentences. The dataset contains sentences very different in wording with a similar meaning as well as sentences similar on the surface level but very different in meaning.

However, the dataset has certain limitations. For instance, it contains several generalizations of a sentence but their mutual relations are no further studied. In other words, we do not know, which ones are more general and should be embedded closer to the original sentence. Our work directly builds upon [7]. We decided to make the dataset more robust by extending it with more sentences and also to ensure that sentences are related to each other whenever possible. We also created a tool to automatically evaluate the quality of embeddings using our dataset and used it to compare several off-the-shelf Czech embeddings.

2.2 Sentence Embedding Methods

Since we extend the Costra dataset, we stick to the Czech language. Our goal is to test as many off-the-shelf Czech sentence embeddings as possible. Unfortunately, to our best knowledge there is only one directly learned representation for entire sentences available for the Czech language: LASER [3].

However, there are available pretrained language models such as multilingual BERT (mBERT) [10] or Flair [2]. Despite neglecting the word order, these methods yield surprisingly strong results in many downstream tasks. In order to move from word vector representations towards representations for entire sentences, we simply average embeddings of hidden states of all tokens in a sentence. For BERT, we also consider the CLS token as a sentence embedding.

Sentence multilingual BERT[1] (SentBERT) is a sentence encoder initialized with multilingual BERT and fine-tuned using MultiNLI [19] and XNLI [9] datasets. The recommended sentence representations are mean-pooled token embeddings, we use the CLS token too.

3 Annotation

We acquired the data in two rounds of annotation. In the first one, we were concentrating on adding more related sentences, i.e., making the sentence space denser. In order to project sentence transformation to a linear scale, we decided to collect *interpolations* and *extrapolations*. In the second round, we collected pairwise comparisons of sentences from both Costra 1.0 and our first round.

3.1 First Round: Collecting Interpolations and Extrapolations

In the first round of annotation, we present annotators with a seed sentence and its transformation and ask them to write the following two new sentences: **interpolation** – a sentence with meaning/style between the two sentences, and **extrapolation** – a sentence with meaning/style even further away from the seed sentence than the transformation in the suggested direction. An example of one annotation is presented in Figure 1.

From the 14 transformation types available in Costra 1.0, we did not select all types of transformation for the first round.[2] The reason was straightforward: it does not make sense to collect interpolations or extrapolations for some of them. For example, meaning of *paraphrases* should be identical or very close to original sentences and searching for interpolation would be a waste of annotators' time. Similarly, there is the *non-sense* transformation, which is created by shuffling content words of a seed sentence, so the final sentence is grammatically correct but has no meaning. There are no interpolations or extrapolations of nonsense.

We manually examined all transformation types and selected only 6 of them that look most linearly scalable: *formal sentence*, *future*, *generalization*, *nonstandard sentence*, *opposite meaning* and *past*. We do not introduce any new type of transformations.

We collected almost 1,500 annotations from 7 annotators, containing 2,749 unique sentences. Total volume of Costra 1.1 is 6,968 sentences.

Implied Sentence Comparison In the second round of annotations, the annotators are sorting sentence pairs. We however know that an interpolation is closer in meaning or style to the seed sentence than its pre-existing transformation or the extrapolation. These implied relations provide us with almost 7,000 sentence comparisons.

[1] http://docs.deeppavlov.ai/en/master/features/models/bert.html

[2] Costra 1.0 contains the following 14 different transformation types: *paraphrase*, *different meaning*, *opposite meaning*, *nonsense*, *minimal change*, *generalization*, *gossip*, *formal sentence*, *non-standard sentence*, *simple sentence*, *possibility*, *ban*, *future*, *past*.

seed	"*Občas se mi na hlavě málo prokrvuje kůže.*" The skin on my head sometimes fills with little blood.
interpolation	"*Kůže na hlavě se mi prokrvuje tak akorát*" The skin on my head fills with just the right amount of blood.
transformation	"*Občas se mi na hlavě hodně prokrvuje kůže*" The skin on my head sometimes fills with too much blood.
extrapolation	"*Nemám žádnou kůži na hlavě*" There is no skin on my head.

Fig. 1. Example from the first round of annotations. The annotator filled the interpolation and extrapolation to the seed and its transformation with *opposite meaning.*

3.2 Second Round: Sentence Comparison

Again, we have manually chosen transformation categories to be compared. We selected those that are linearly comparable, i.e. changes in tense (*future*, *past*), changes in style (*formal sentence*, *gossip*, *nonstandard sentence*, *simple sentence*) and significant changes in meaning (*generalization*, *opposite meaning*). We merged two categories (*non-standard* and *gossip*) because the actual sentences in the collection often realized 'gossipping' via non-standard language and vice versa.

The annotators were presented with a pair of sentences and criteria, how to compare them.[3] Of course, not always are the sentences comparable. Their meaning might be either very close or very far from each other, both making them hard to compare. For every pair of sentences S_1 and S_2, the annotators had the following four options:

1. S_1 is more general/formal/in the past/non-standard/... than S_2.
2. S_2 is more general/formal/in the past/non-standard/... than S_1
3. S_1 and S_2 are **too similar**, for example: "*Byl rozčilený a hodně mluvil.*" (He was upset and talked a lot.) and "*Ovlivněn silnými emocemi říkal ledacos.*" (Influenced by strong emotions, he said all kind of things.) are so close in their meaning that it is almost impossible to select the more general one.
4. S_1 and S_2 **too dissimilar**, for example: neither of the sentences "*Všechno zlé je pro něco dobré.*" (Every cloud has a silver lining; lit. All bad is good for something.) and "*V Asii jsou různá období.*" (There are different seasons in Asia.) is generalization of the other sentence, even though they both were created as generalizations of the sentence "*Bangladéšská monzunová sezóna přináší radost, problémy i pozoruhodné fotografie*" (Bangladesh's monsoon season brings joy, problems and remarkable photos.)

We collected more than 25k sentence pairwise comparisons from 7 annotators. We compute inter and intra-annotator agreement using average pairwise Kohen's

[3] Only for *opposite meaning* the annotators were presented with three sentences: two candidates and a source sentence. The annotators were then supposed to say which of the candidates is closer to meaning of the source sentence.

kappa [8]. The scores are generally good, not lower than other types of linguistic annotation. Our inter-annotator agreement is 0.62 ($\kappa = 0.49$) and our intra-annotator agreement is 0.77 ($\kappa = 0.7$).

4 Vector Evaluation

4.1 Sentence Comparison

We combine sentence comparisons obtained in the first and second round of the annotation. A pair of sentences can have multiple annotations in the collection. We trust the annotation only if there is an option with the majority of votes.

We keep 16,385 sentence pairs with human comparison and 1,620 were disregarded because of a disagreement in annotators' judgments.

4.2 Sentence Evaluation

We evaluate sentence embeddings in 12 scales grouped into 6 classes for conciseness. Two focus on transformations without an assumed linear scale behind: **basic**: paraphrases should be closer to their seed than any transformation, which significantly changed the meaning of the seed (*different meaning*, *nonsense*, *minimal change*), **modality**: paraphrases should be closer to their seed than any transformation, which changes modality of the seed (*possibility*, *ban*).

The remaining four classes evaluate whether sentence space reflects the ordering implied by the collected comparisons: **time** (how often the mutual ordering of all transformation towards *future* matches the relative distances in the embedding space; similarly but separately for *past*), **style** (*formal sentence*, *non-standard sentence*, *simple sentence*), **generalization**, and **opposite**.

For categories in the first two classes, we compute the accuracy of sentence embeddings, i.e., how often $\mathrm{sim}_C(\mathrm{v}_{\mathrm{seed}}, \mathrm{v}_{\mathrm{P}}) > \mathrm{sim}_C(\mathrm{v}_{\mathrm{seed}}, \mathrm{v}_{\mathrm{T}})$ for every paraphrase P and every transformation T of the particular category and in the examined sentence embeddings $\mathrm{v}_{\bullet}$.

For categories in the latter four classes, the evaluation is based on collected judgments. So if the annotators judge that sentences A, B and C satisfy A<B and B<C, we test how often $\mathrm{sim}_C(\mathrm{v}_{\mathrm{A}}, \mathrm{v}_{\mathrm{B}}) > \mathrm{sim}_C(\mathrm{v}_{\mathrm{A}}, \mathrm{v}_{\mathrm{C}})$ and $\mathrm{sim}_C(\mathrm{v}_{\mathrm{B}}, \mathrm{v}_{\mathrm{C}}) > \mathrm{sim}_C(\mathrm{v}_{\mathrm{A}}, \mathrm{v}_{\mathrm{C}})$ To make use of the options *too similar* and *too dissimilar*, we check whether $\mathrm{sim}_C(\mathrm{v}_{\mathrm{A}}, \mathrm{v}_{\mathrm{B}}) > \mathrm{sim}_C(\mathrm{v}_{\mathrm{B}}, \mathrm{v}_{\mathrm{C}})$ for all sentences A, B, C where the annotations indicate that A and B are too similar to each other and B and C are too dissimilar.

5 Results

As Table 1 shows, none of the examined sentence embeddings are particularly good in the basic requirement of paraphrases being embedded closer to each other than sentences with a significantly different meaning. The best performing method mBERT-CLS reach the accuracy of 26%. This contrasts with [13],

Table 1. Experimental results: Geometric relations in sentence embedding spaces

	basic	modality	time	style	gener.	opposite	avg
SentBERT - mean	0.150	0.251	0.667	0.588	**0.718**	0.685	0.510
SentBERT - CLS	0.172	**0.303**	0.654	0.577	0.690	0.654	0.508
Flair - mean	0.145	0.157	**0.682**	**0.627**	0.695	**0.728**	0.506
mBERT - CLS	**0.262**	0.274	0.616	0.579	0.603	0.640	0.496
mBERT - mean	0.103	0.115	0.674	0.621	0.691	0.727	0.489
LASER	0.255	0.244	0.583	0.533	0.667	0.636	0.486

which shows that LASER is particularly good at identifying related sentences in Polish. However, we must emphasize that transformations in the **basic** class were purposefully selected to pose a difficult challenge[4] – only very sophisticated embedding method can achieve high accuracy, which is precisely the purpose of this testing dataset.

As one can expect, the first two tasks turned out too hard for all BOW embeddings that use mean to calculate the final vector. On the other hand, LASER and mBert-CLS perform surprisingly well with more than one-fourth of paraphrases embedded close to their seeds.

The linearity of time, style, level of generality or the level of opposition are reflected considerably better: 63–74% of tested sentence triples satisfy the expectation. Mean-based embeddings (Flair and mBert in particular) achieve the best performance in this evaluation of linear relations.

6 Conclusion

We presented an extension of COSTRA 1.0, a corpus of sentence transformations, providing new transformations and relations in order to examine to what extent embedding spaces reflect linear ordering with regard to certain kinds of sentence relationships.

We find that paraphrases are often embedded too far from each other and many meaning-altering transformations lie in a closer range. This confirms that the selected transformations are not easy to capture since all BOW methods perform very poorly on them. The natural ordering of sentences with respect to time, style and level of generalization or opposition is embedded considerably better.

Interestingly, the only directly learned sentence embedding LASER shows on average the worst results from all tested methods. However, the differences between all methods are very small.

Our hope is that Costra 1.1 will help to develop new better sentence embedding for the Czech language. It is freely available at the following link:

[4] *Different meaning*, *nonsense* and *minimal change* are all very similar in wording to a seed sentence unlike its paraphrases, which must use different words to express similar meaning. For more details see [7]

http://hdl.handle.net/11234/1-3248

Easy-to-use Czech sentence embeddings quality evaluator is available here:

https://github.com/barancik/costra

References

1. Adi, Y., et al.: Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. CoRR **abs/1608.04207** (2016)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: COLING (2018)
3. Artetxe, M., Schwenk, H.: Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. CoRR **abs/1812.10464** (2018)
4. Bacon, G., Regier, T.: Probing sentence embeddings for structure-dependent tense. In: EMNLP BlackboxNLP (2018)
5. Bakarov, A.: A survey of word embeddings evaluation methods. CoRR **abs/1801.09536** (2018)
6. Barančíková, P., Bojar, O.: In search for linear relations in sentence embedding spaces. In: ITAT SloNLP (2019)
7. Barančíková, P., Bojar, O.: COSTRA 1.0: A dataset of complex sentence transformations. In: LREC (2020)
8. Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics **22**(2) (1996)
9. Conneau, A., et al.: XNLI: Evaluating cross-lingual sentence representations. In: EMNLP (2018)
10. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
11. Ettinger, A., Elgohary, A., Resnik, P.: Probing for semantic evidence of composition by means of simple classification tasks. In: ACL RepEval (2016)
12. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: NIPS (2015)
13. Krasnowska-Kieraś, K., Wróblewska, A.: Empirical linguistic study of sentence embeddings. In: ACL (2019)
14. Linzen, T., Dupoux, E., Goldberg, Y.: Assessing the ability of lstms to learn syntax-sensitive dependencies. TACL **4** (2016)
15. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space (2013)
16. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL/HLT (2013)
17. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: EMNLP (2015)
18. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. CoRR **abs/1511.08198** (2015)
19. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. In: NAACL-HLT (2018)
20. Zhu, X., Li, T., de Melo, G.: Exploring semantic properties of sentence embeddings. In: ACL (2018)
21. Zhu, X., de Melo, G.: Sentence analogies: Exploring linguistic relationships and regularities in sentence embeddings (2020)