In Search for Linear Relations in Sentence Embedding Spaces

Petra Barančíková, Ondřej Bojar

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{barancikova,bojar}@ufal.mff.cuni.cz

Abstract: We present an introductory investigation into continuous-space vector representations of sentences. We acquire pairs of very similar sentences differing only by a small alterations (such as change of a noun, adding an adjective, noun or punctuation) from datasets for natural language inference using a simple pattern method. We look into how such a small change within the sentence text affects its representation in the continuous space and how such alterations are reflected by some of the popular sentence embedding models. We found that vector differences of some embeddings actually reflect small changes within a sentence.

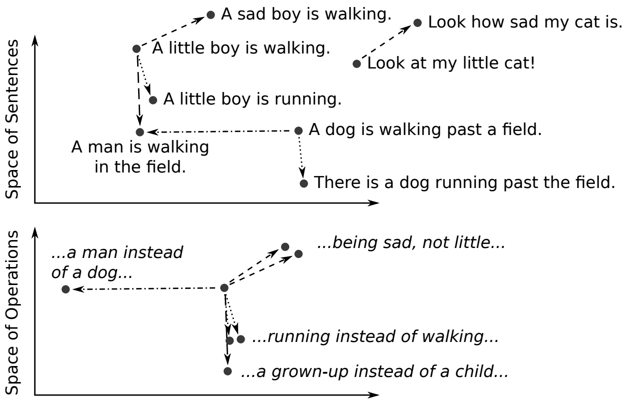
1 Introduction

Continuous-space representations of sentences, so-called sentence embeddings, are becoming an interesting object of study, consider e.g. the BlackBox workshop.[1] Representing sentences in a continuous space, i.e. commonly with a long vector of real numbers, can be useful in multiple ways, analogous to continuous word representations (word embeddings). Word embeddings have provably made downstream processing robust to unimportant input variations or minor errors (sometimes incl. typos), they have greatly boosted the performance of many tasks in low data conditions and can form the basis of empirically-driven lexicographic explanations of word meanings.

One notable observation was made in [15], showing that several interesting relations between words have their immediate geometric counterpart in the continuous vector space.

Our aim is to examine existing continuous representations of *whole sentences*, looking for an analogous behaviour. The idea of what we are hoping for is illustrated in Figure 1. As with words, we would like to learn if and to what extent some simple geometric operations in the continuous space correspond to simple semantic operations on the sentence strings. Similarly to [15], we are deliberately *not including* this aspect in the training objective of the sentence presentations but instead search for properties that are learned in an unsupervised way, as a side-effect of the original training objective, data and setup.

Figure 1: An illustration of a continuous multi-dimensional vector space representing individual sentences, a ‘space of sentences’ (upper plot) where each sentence is represented as a dot. Pairs of related sentences are connected with arrows; dashing indicates various relation types. The lower plot illustrates a possible ‘space of operations’ (here vector difference, so all arrows are simply moved to start at a common origin). The hope is that similar operations (e.g. all vector transformations extracted from sentence pairs differing in the speed of travel “running instead of walking”) would be represented close to each other in the space of operations, i.e. form a more or less compact *cluster*.


Space of Sentences
A sad boy is walking.
Look how sad my cat is.
A little boy is walking.
Look at my little cat!
A little boy is running.
A dog is walking past a field.
A man is walking in the field.
There is a dog running past the field.
Space of Operations
...a man instead of a dog...
...being sad, not little...
...running instead of walking...
...a grown-up instead of a child...

This approach has the potential of *explaining* the good or bad performance of the examined types of representations in various tasks.

The paper is structured as follows: Section 2 reviews the closest related work. Sections 3 and 4, respectively, describe the dataset of sentences and the sentence embeddings methods we use. Section 5 presents the selection of operations on the sentence vectors. Section 6 provides the main experimental results of our work. We conclude in Section 7.

2 Related Work

Series of tests to measure how well their word embeddings capture semantic and syntactic information is defined in [15]. These tests include for example declination of adjectives (“easy"→“easier"→“easiest"), chang-

[1] https://blackboxnlp.github.io/

Figure 2: Example of our pattern extraction method. In the first step, the longest common subsequence of tokens (*ear is playing a guitar .*) is found and replaced with the variable X. In the second step, *with a tattoo behind* is substituted with the variable Y. As the variables are not listed alphabetically in the premise, they are switched in the last step.

step	premise	hypothesis
1.	a man with a tattoo behind his **ear is playing a guitar .**	a woman with a tattoo behind her **ear is playing a guitar .**
2.	a man **with a tattoo behind** his X	a woman **with a tattoo behind** her X
3.	a man Y his X	a woman Y her X
4.	a man X his Y	a woman X her Y

Figure 3: Top 10 patterns extracted from sentence pairs labelled as entailmens, contradictions and neutrals, respectively. Note the “X → X" pattern indicating no change in the sentence string at all.

	entailments			**contradictions**			**neutrals**		
	premise	hypothesis		premise	hypothesis		premise	hypothesis	
1.	X	X	693	X man Y	X woman Y	413	X Y	X sad Y	701
2.	X man Y	X person Y	224	X woman Y	X man Y	196	X Y	X big Y	119
3.	X .	X	207	X men	X women Y	111	X Y	X fat Y	69
4.	X woman Y	X person Y	118	X boy Y	X girl Y	109	X young Y	X sad Y	68
5.	X boy Y	X person Y	65	X dog Y	X cat Y	98	X people Y	X men Y	60
6.	X Y	Y , X .	61	X girl Y	X boy Y	97	X	sad X	51
7.	X men Y	X people Y	56	X women Y	X men Y	64	X	X	41
8.	two X	X	56	X Y,	X not Y	56	X person Y	X man Y	34
9.	X girl Y	X person Y	55	two X,	three X	46	X Y	X red Y	30
10.	X , Y	Y X .	53	X child Y	X man Y	44	X Y	X busy Y	28

ing the tense of a verb (“walking"→“walk") or getting the capital (“Athens"→“Greece") or currency of a state (“Angola"→“kwanza"). References [2; 13] have further refined the support of sub-word units, leading to considerable improvements in representing morpho-syntactic properties of words. Vylomova, Rimmel, Cohn and Baldwin [26] largely extended the set of considered semantic relations of words.

Sentence embeddings are most commonly evaluated extrinsically in so called ‘transfer tasks’, i.e. comparing the evaluated representations based on their performance in sentence sentiment analysis, question type prediction, natural language inference and other assignments. Reference [8] introduce ‘probing tasks’ for intrinsic evaluation of sentence embeddings. They measure to what extent linguistic features like sentence length, word order, or the depth of the syntactic tree are available in a sentence embedding. This work was extended to SentEval [6], a toolkit for evaluating the quality of sentence embedding both intrinsically and extrinsically. It contains 17 transfer tasks and 10 probing tasks. SentEval is applied to many recent sentence embedding techniques showing that no method had a consistently good performance across all tasks [18].

Voleti, Liss and Berisha [25] examine how errors (such as incorrect word substitution caused by automatic speech recognition) in a sentence affect its embedding. The embeddings of corrupted sentences are then used in textual similarity tasks and the performance is compared with original embedding. The results suggest that pretrained neural sentence encoders are much more robust to introduced errors contrary to bag-of-words embeddings.

3 Examined Sentences

Because manual creation of sentence variations is costly, we reuse existing data from SNLI [3] and MultiNLI [27]. Both these collections consist of pairs of sentences—a premise and a hypothesis—and their relationship (entailment/contradiction/neutral). The two datasets together contain 982k unique sentence pairs. All sentences were lowercased and tokenized using NLTK [14].

From all the available sentence pairs, we select only a subset where the difference between the sentences in the pair can be described with a simple pattern. Our method goes as follows: given two sentences, a premise p and the corresponding hypothesis h, we find the longest common substring consisting of whole words and replace it with a variable. This is repeated once more, so our sentence patterns can have up to two variables. In the last step, we make sure the pattern is in a canonical form by switching the variables to ensure they are alphabetically sorted in p. The process is illustrated in Figure 2.

Ten most common patterns for each NLI relation are shown in Figure 3. Many of the obtained patterns clearly match the sentence pair label. For instance the pattern no. 2 (“X man Y → X person Y”) can be expected to lead to a sentence pair illustrating entailment. If a man appears in a story, we can infer that a person appeared in the story. The contradictions illustrate typical oppositions like man–woman, dog–cat. Neutrals are various refinements of the content described by the sentences, probably in part due to the original instruction in SNLI that hypothesis "might be a true" given the premise in neutral relation.

We kept only patterns appearing with at least 20 different sentence pairs in order to have large and variable sets of sentence pairs in subsequent experiments. We also ignored the overall most common pattern, namely the identity, because it actually does not alter the sentence at all. Strangely enough, identity was observed not just among entailment pairs (693 cases), but also in neutral (41 cases) and contradiction (22) pairs.

Altogether, we collected 4,2k unique sentence pairs in 60 patterns. Only 10% of this data comes from MultiNLI, the majority is from SNLI.

4 Sentence Embeddings

We experiment with several popular pretrained sentence embeddings.

InferSent[2] [7] is the first embedding model that used a supervised learning to compute sentence representations. It was trained to predict inference labels on the SNLI dataset. The authors tested 7 different architectures and BiLSTM encoder with max pooling achieved the best results. InferSent comes in two versions: **InferSent_1** is trained with Glove embeddings [17] and **InferSent_2** with fastText [2]. InferSent representations are by far the largest, with the dimensionality of 4096 in both versions.

Similarly to InferSent, Universal Sentence Encoder [4] uses unsupervised learning augmented with training on supervised data from SNLI. There are two models available. **USE_T**[3] is a transformer-network [23] designed for higher accuracy at the cost of larger memory use and computational time. **USE_D**[4] is a deep averaging network [12], where words and bi-grams embeddings are averaged and used as input to a deep neural network that computes the final sentence embeddings. This second model is faster and more efficient but its accuracy is lower. Both models output representation with 512 dimensions.

Unlike the previous models, **BERT**[5] (Bidirectional Encoder Representations from Transformers) [10] is a deep unsupervised language representation, pre-trained using only unlabeled text. It has two self-supervised training objectives - masked language modelling and next sentence classification. It is considered bidirectional as the Transformer encoder reads the entire sequence of words at once. We use a pre-trained BERT-Large model with Whole Word Masking. BERT gives embeddings for every (sub)word unit, we take as a sentence embedding a [CLS] token, which is inserted at the beginning of every sentence. BERT embeddings have 1,024-dimensions.

ELMo[6] (Embedding from Language Models) [5] uses representations from a biLSTM that is trained with the language model objective on a large text dataset. Its embeddings are a function of the internal layers of the bi-directional Language Model (biLM), which should capture not only semantics and syntax, but also different meanings a word can represent in different contexts (polysemy). Similarly to BERT, each token representation of ELMo is a function of the entire input sentence - one word gets different embeddings in different contexts. ELMo computes an embedding for every token and we compute the final sentence embedding as the average over all tokens. It has dimensionality 1024.

LASER[7] (Language-Agnostic SEntence Representations) [1] is a five-layer bi-directional LSTM (BiLSTM) network. The 1,024-dimension vectors are obtained by max-pooling over its last states. It was trained to translate from more than 90 languages to English or Spanish at the same time, the source language was selected randomly in each batch.

5 Choosing Vector Operations

Mikolov, Chen, Corrado and Dean [15] used a simple vector difference as the operation that relates two word embeddings. For sentences embeddings, we experiment a little and consider four simple operations: addition, subtraction, multiplication and division, all applied elementwise. More operations could be also considered as long as they are reversible, so that we can isolate the vector change for a particular sentence alternation and apply it to the embedding of any other sentence. Hopefully, we would then land in the area where the correspondingly altered sentence is embedded.

The underlying idea of our analysis was already sketched in Figure 1. From every sentence pair in our dataset, we extract the pattern, i.e. the string edit of the sentences. The arithmetic operation needed to move from the embedding of the first sentence to the embedding of the second sentence (in the continuous space of sentences) can be represented as a point in what we call the *space of operations*. Considering all sentence pairs that share the same edit pattern, we obtain many points in the space of operations. If the space of sentences reflects the particular edit pattern in an accessible way, all the corresponding points in the space of operations will be close together, forming a cluster.

To select which of the arithmetic operations best suits the data, we test pattern clustering with three common clustering performance evaluation methods:

[2] https://github.com/facebookresearch/InferSent

[3] https://tfhub.dev/google/universal-sentence-encoder-large/3

[4] https://tfhub.dev/google/universal-sentence-encoder/2

[5] https://github.com/google-research/bert

[6] https://github.com/HIT-SCIR/ELMoForManyLangs

[7] https://github.com/facebookresearch/LASER

Table 1: This table presents the quality of pattern clustering in terms of the three cluster evaluation measures in the space of operations. For all the scores, the value of 1 represents a perfect assignment and 0 corresponds to random label assignment. All the numbers were computed using the Scikit-learn library [16]. Best operation according to each cluster score across the various embeddings in bold.

		Adjusted Rank Index				V-measure				Adjusted Mutual Information			
embedding	dim.	-	+	*	/	-	+	*	/	-	+	*	/
InferSent_1	4096	**0.58**	0.03	0.03	0.00	**0.91**	0.28	0.24	0.03	**0.87**	0.18	0.14	0.00
ELMo	1024	0.55	0.03	0.02	0.00	0.85	0.28	0.23	0.03	0.82	0.18	0.13	0.00
LASER	1024	0.48	0.02	0.01	0.00	0.79	0.19	0.15	0.03	0.76	0.09	0.04	0.00
USE_T	512	0.25	0.04	0.08	0.00	0.73	0.25	0.30	0.03	0.69	0.14	0.20	0.00
InferSent_2	4096	0.31	0.04	0.04	0.01	0.69	0.28	0.28	0.10	0.65	0.19	0.19	0.03
BERT	1024	0.33	0.02	0.01	0.00	0.66	0.22	0.16	0.03	0.62	0.12	0.06	0.00
USE_D	512	0.21	0.05	0.08	0.00	0.65	0.27	0.33	0.03	0.58	0.17	0.23	0.00
average	1775	**0.39**	0.03	0.04	0.00	**0.75**	0.25	0.24	0.04	**0.71**	0.15	0.14	0.00

- **Adjusted Rand index** [11] is measure of the similarity between two cluster assignments adjusted with chance normalization. The score ranges from -1 to +1 with 1 being the perfect match score and values around 0 meaning random label assignment. Negative numbers show worse agreement than what is expected from a random result.

- **V-measure** [19] is harmonic mean of *homogeneity* (each cluster should contain only members of one class) and *completeness* (all members of one class should be assigned to the same cluster). The score ranges from 0 (the worst situation) to 1 (perfect score).

- **Adjusted Mutual Information** [21] measures the agreement of the two clusterings with the correction of agreement by chance. The random label assignment gets a score close to 0, while two identical clusterings get the score of 1.

As the detailed description of these measures is out of scope of this article, we refer readers to related literature (e.g. [24]). We use these scores to compare patterns with labels predicted by k-Means (best result of 100 random initialisations). The results are presented in Table 1. It is apparent that the best distribution by far is achieved using the most intuitive operation, vector subtraction.

There seems to be a weak correlation between the size of embeddings and the scores. The smallest embeddings USE_D and USE_T are getting the worst scores, while the largest embeddings InferSent_1 are the best scoring embeddings. However, InferSent_2 with dimensionality 4096 is performing poorly. The fact that several of the embeddings were trained on SNLI does not to seem benefit those embeddings. Between the three top scored embeddings, only InferSent_1 was trained on the data that we use for evaluation of embeddings.

6 Experiments

For the following exploration of the continuous space of operations, we focus only on the ELMo embeddings. They scored second best in all scores but unlike the best scoring Infersent_1, ELMo was not trained on SNLI, which is the major source of our sentence pairs.

The t-SNE [22] visualisation of subtractions of ELMo vectors is presented in Figure 4. The visualisation is constructed automatically and, of course, *without* the knowledge of the pattern label. It shows that the patterns are generally grouped together into compact clusters with the exception of a ‘chaos cloud’ in the middle and several outliers. Also there are several patterns that seem inseparable, e.g. “two X → X" and “three X → X", or “X white Y -> X Y" and “X black Y -> X Y".

We identified the patterns responsible for the noisy center and outliers by computing weighted inertia for each pattern (the sum of squared distances of samples to their cluster center divided by the size of sample). The clusters with highest inertia consists of patterns representing a change of word order and/or adding or removing punctuation. These patterns are:

X is Y . → Y is X	X Y . → Y X .	X → X .
X , Y . → Y X .	X , Y . → Y , X .	
X Y . → Y , X .	X . → X	

To see if the space of operations can be interpreted also automatically, i.e. if the sentence relations are generalizable, we remove the noisy patterns as above and apply fully unsupervised clustering: we do not even disclose the expected number of patterns, i.e. clusters. We try two metrics for finding the optimal number of clusters: Davies-Bouldin’s index [9] and Silhouette Coefficient [20]. They are both designed to measure compactness and separation of the clusters, i.e. they award dense clusters that are far from each other. Both Davies-Bouldin index and Silhouette Coefficient agree that the best separation is achieved

Figure 4: t-SNE representation of patterns. The points in the operation space are obtained by subtracting the ELMo embedding of the hypothesis from the ELMo embedding of the premise. Best viewed in color. Colors correspond to the sentence patterns.

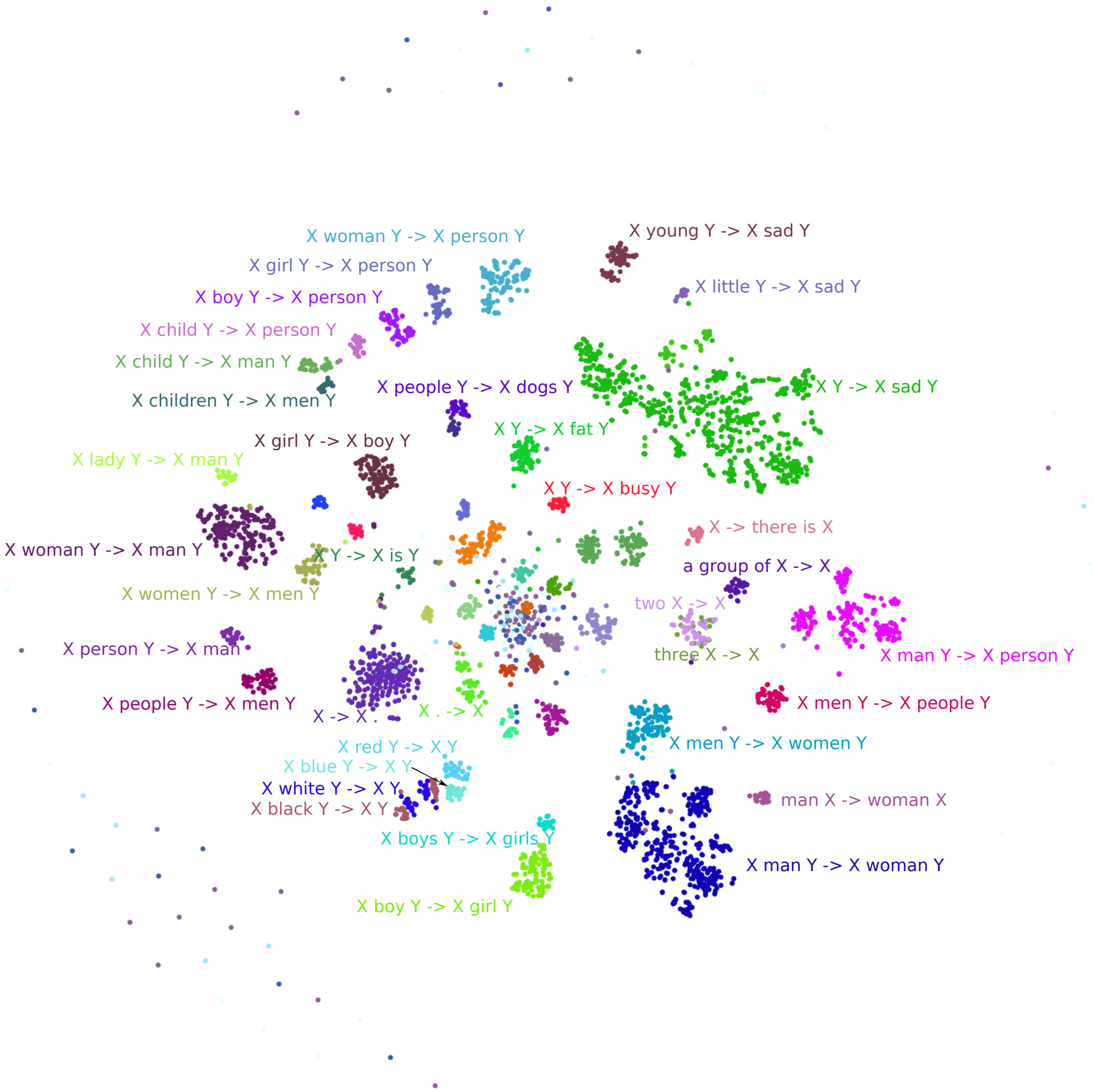
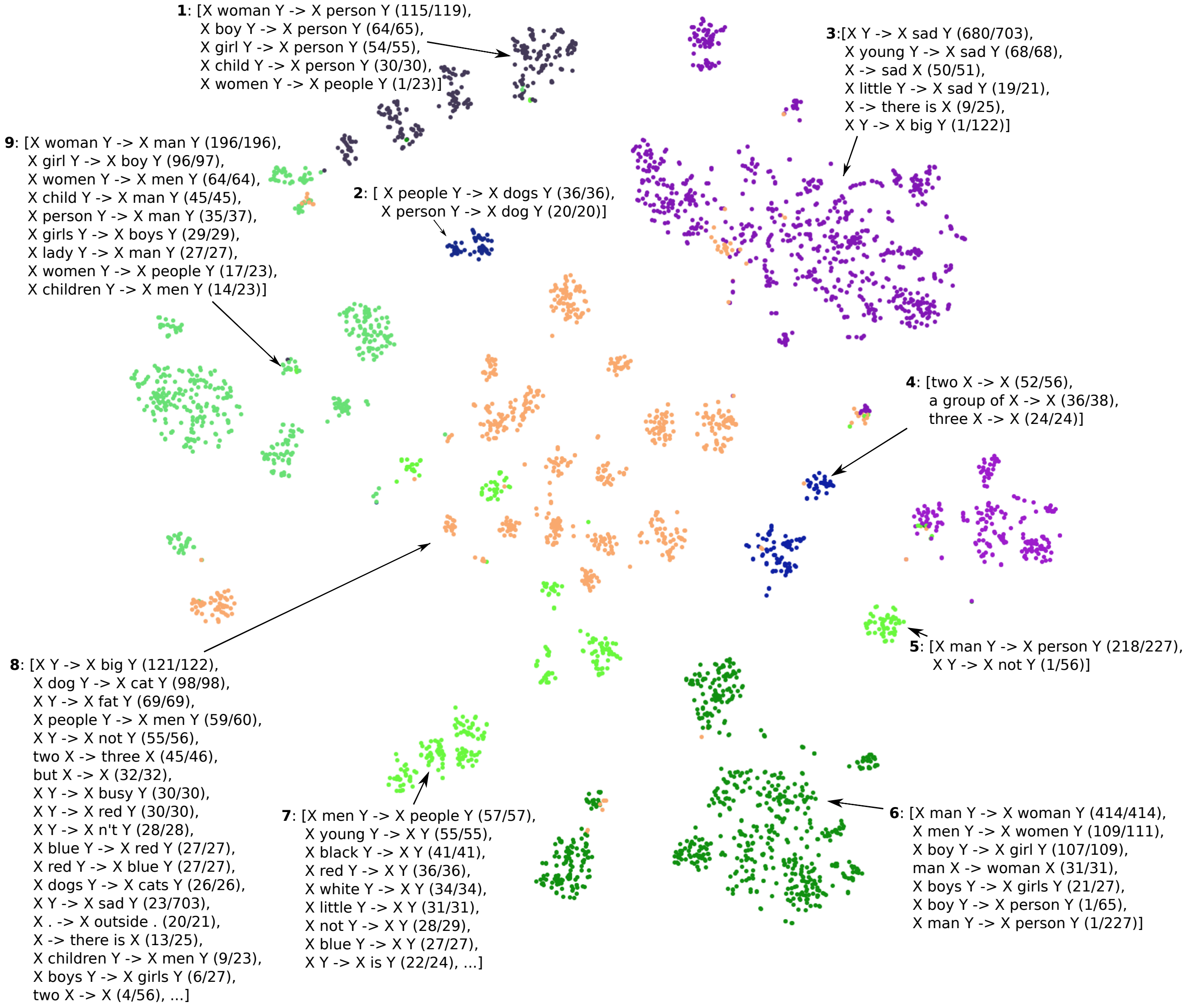


Figure 5: t-SNE representation of patterns as in Figure 4 with colors coding now fully automatic clusters. Each cluster is labelled with the set of patterns extracted from sentence pairs assigned to the cluster. The numbers in parentheses indicate how many sentence pairs belong to the given pattern within this cluster and overall, resp. For instance the line “two X → X (52/56)” says that of the 56 sentence pairs differing in the prefix “two”, 52 were automatically clustered together based on the subtraction of their ELMo embeddings.



1: [X woman Y -> X person Y (115/119),
X boy Y -> X person Y (64/65),
X girl Y -> X person Y (54/55),
X child Y -> X person Y (30/30),
X women Y -> X people Y (1/23)]

3:[X Y -> X sad Y (680/703),
X young Y -> X sad Y (68/68),
X -> sad X (50/51),
X little Y -> X sad Y (19/21),
X -> there is X (9/25),
X Y -> X big Y (1/122)]

9: [X woman Y -> X man Y (196/196),
X girl Y -> X boy Y (96/97),
X women Y -> X men Y (64/64),
X child Y -> X man Y (45/45),
X person Y -> X man Y (35/37),
X girls Y -> X boys Y (29/29),
X lady Y -> X man Y (27/27),
X women Y -> X people Y (17/23),
X children Y -> X men Y (14/23)]

2: [X people Y -> X dogs Y (36/36),
X person Y -> X dog Y (20/20)]

4: [two X -> X (52/56),
a group of X -> X (36/38),
three X -> X (24/24)]

5: [X man Y -> X person Y (218/227),
X Y -> X not Y (1/56)]

8: [X Y -> X big Y (121/122),
X dog Y -> X cat Y (98/98),
X Y -> X fat Y (69/69),
X people Y -> X men Y (59/60),
X Y -> X not Y (55/56),
two X -> three X (45/46),
but X -> X (32/32),
X Y -> X busy Y (30/30),
X Y -> X red Y (30/30),
X Y -> X n't Y (28/28),
X blue Y -> X red Y (27/27),
X red Y -> X blue Y (27/27),
X dogs Y -> X cats Y (26/26),
X Y -> X sad Y (23/703),
X . -> X outside . (20/21),
X -> there is X (13/25),
X children Y -> X men Y (9/23),
X boys Y -> X girls Y (6/27),
two X -> X (4/56), ...]

7: [X men Y -> X people Y (57/57),
X young Y -> X Y (55/55),
X black Y -> X Y (41/41),
X red Y -> X Y (36/36),
X white Y -> X Y (34/34),
X little Y -> X Y (31/31),
X not Y -> X Y (28/29),
X blue Y -> X Y (27/27),
X Y -> X is Y (22/24), ...]

6: [X man Y -> X woman Y (414/414),
X men Y -> X women Y (109/111),
X boy Y -> X girl Y (107/109),
man X -> woman X (31/31),
X boys Y -> X girls Y (21/27),
X boy Y -> X person Y (1/65),
X man Y -> X person Y (1/227)]

at 9 clusters. Running k-Means with 9 clusters, we get the result as plotted in Figure 5.

Manually inspecting the contents of the automatically identified clusters, we see that many clusters are meaningful in some way. For instance, Cluster 1 captures 90% (altogether 264 out of 292) sentence pairs exerting the pattern of generalizing women, boys or girls to people. The counterpart for men belonging to people is spread into Cluster 5 (218 out of 227 pairs) for the singular case and not so clean Cluster 7 containing 57/57 of the plural pairs "X men Y → X people Y" together with various oppositions. Cluster 2 covers all sentence pairs where a person is replaced with a dog. Cluster 3 is primarily connected with sentence pairs introducing bad mood. Cluster 4 unites patterns that represent omitting a numeral/group. Cluster 6 covers gender oppositions in one direction and Cluster 9 adds the other direction (with some noise for child/man and person/man and similar), etc.

7 Conclusion and Future Work

We examined vector spaces of sentence representations as inferred automatically by sentence embedding methods such as InferSent or ELMo. Our goal was to find out if some simple arithmetic operations in the vector space correspond to meaningful edit operations on the sentence strings.

Our first explorations of 60 sentence edit patterns document that this is indeed the case. Automatically identified frequent patterns with 20 or more occurrences in the SNLI and MultiNLI datasets correspond to simple vector differences. The ELMo space (and others such as Infersent_1, LASER and USE-T, which are omitted due to paper length requirements) exerts this property very well.

Unfortunately, choosing ELMo as example might not have been the best option – we compute ELMo embeddings by averaging contextualized word embeddings and majority of the patterns are just removing/adding/changing a single word. Difference between two such sentence embeddings may be a simple difference between the embeddings of the words substituted, depending on the effect of the contextualization. Thus, the differences in vector space would show rather the word embeddings than the sentence embeddings.

It should be noted that our search made use of only about 0.5% of the sentence pairs available in SNLI and MultiNLI. The remaining sentence pairs differ beyond what was extractable automatically using our simple pattern method. A different approach for a fine-grained description of the semantic relation between two sentences would have to be taken for a better exploitation of the available data.

Our plans for the long term are to further verify these observations using a more diverse set of vector operations and a larger set of sentence alternations, primarily by extending the set of alternation *types*. We also plan to examine the possibilities of *generating sentence strings* back from the sentence embedding space. If successful, our method could lead to controlled paraphrasing via the continuous space: take an input sentence, embed it, modify the embedding using a vector operation and generate the target sentence in the standard textual from.

Acknowledgment

This work has been supported by the grant No. 18-24210S of the Czech Science Foundation. It has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References

[1] M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464, 2018.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[4] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[5] W. Che, Y. Liu, Y. Wang, B. Zheng, and T. Liu. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *CoRR*, abs/1807.03121, 2018.

[6] A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

[7] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364, 2017.

[8] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *CoRR*, abs/1805.01070, 2018.

[9] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, Feb. 1979.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[11] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, Dec 1985.

[12] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[13] T. Kocmi and O. Bojar. Subgram: Extending skip-gram word representation with substrings. *CoRR*, abs/1806.06571, 2018.

[14] E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[15] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

[18] C. S. Perone, R. Silveira, and T. S. Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *CoRR*, abs/1806.06259, 2018.

[19] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[20] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, Nov. 1987.

[21] A. Strehl and J. Ghosh. Cluster ensembles: A knowledge reuse framework for combining partitionings. In *Eighteenth National Conference on Artificial Intelligence*, pages 93–98, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[22] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[24] N. X. Vinh and J. Epps. A novel approach for automatic number of clusters detection in microarray data based on consensus clustering. In *2009 Ninth IEEE International Conference on Bioinformatics and Bio-Engineering*, pages 84–91, June 2009.

[25] R. Voleti, J. M. Liss, and V. Berisha. Investigating the effects of word substitution errors on sentence embeddings. *CoRR*, abs/1811.07021, 2018.

[26] E. Vylomova, L. Rimell, T. Cohn, and T. Baldwin. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *CoRR*, abs/1509.01692, 2015.

[27] A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, 2018.