Exploring dimension reduction techniques for text dataset visualization

1st Dinar Zaiakhov
Machine learning and knowledge representation lab
Innopolis Univesity
Innopolis, Russia
d.zayahov@innopolis.university

2nd Stanislav Protasov
Machine learning and knowledge representation lab
Innopolis University
Innopolis, Russia
s.protasov@innopolis.ru

***Abstract*—Large multidimensional data sets are hard to visualize. Most existing methods dedicate visual space to multiple items or multiple features. In this work, we explore dimensionality reduction methods to capture both properties. We show that self-organizing maps (SOM) are the good choice for screen and paper visualization. We involve colors to make multiple texts comparable on a single image. We discuss important properties of our visualization method and propose an optimal parameter set with respect to text vocabulary size. Our methods are implemented in python programming language and are available as an open-source visualization library.**

***Index Terms*—dimensionality reduction, self-organizing maps, word embeddings, data visualization**

I. Introduction

There exist numerous ways to visualize the data. Some of these methods target large data sets and present them as a whole, e.g., maps or histograms. Scatter plots, box plots, and heat maps capture two or three important data features, and display data points with coordinates and colors. On the other hand, different tabular and similar methods can give a tribune to high-dimensional data. Spectrograms and tables allow us to observe the features presented evenly. The problem arises when the data is both multiple and high-dimensional. We can dedicate the visual space either to data points (maps), or to data features (spectrograms), but not to both. The answer to this challenge usually comes with understanding that raw vector space points are less important than their relations. Bringing the concept of a metric (distance) to the vector data sets allows to concentrate on such relative concepts as *similarity*, *clustering*, and *neighborhood*, and get rid of absolute representations.

Since 2013 [9] vector representation of words has opened the way for latent space arithmetic. Previously, language terms had only binary pairwise relations (capital to country, adverb to adjective, synonyms). Today we know that these relations can be encoded as a metric of a vector space. Simply saying, with word embedding models we can, for any pair of words, compute a number (distance) which will reflect how similar they are. The direction of this link can also shed some light on the nature of this similarity. In practice (e.g., in information retrieval), we are happy with just the length of this link.

The computer methods of visualization in computer linguistics have evolved for a long time. WordNet [10] graphs were introduced to highlight the previously mentioned *relations*, while word clouds (also known as tag clouds) [3] concentrate on the *importance* of the word in some data sets. In our work, we try to address both features at one shot: how to visualize important concepts and show their place in mutual relations of words. We started our research [14] with the straightforward implementation of dimensionality reduction. Current work follows our previous method and tries to find an optimal (in sense of subjective perception) way for key concept visualization of textual data, where we account both structure and importance. We illustrate our findings with the corpus of the physics texts obtained from `arXiv.org`. We used 10 articles each from 2010, 2017 and 2020 which served for us as a foreground distribution. The background data consists of 32719 words (nouns and verbs) from Corpus of Contemporary American English (COCA) [2]. Results of our work are delivered as an open-source python package[1]. One can also find the test data in our repository.

Here we enumerate the major contributions of the paper:

- We did a comparison of dimension reduction techniques with respect to our problem and chose the method which is the most suitable for text data visualization specifically.
- We proposed a set of recommendations to find the optimal parameters for different text data set sizes.
- We explored text vectorization techniques and chose the method which is the best in embedding relations into metric vector space.
- We made our findings available to the community via open source software.

The paper is organized as follows. In section II we discuss related works on word and text vectorization and dimension reduction methods. Section III is devoted to the proposed experiment methodology. In Section IV we present the experiment results and quantitative observations, while in section V we discuss them from a qualitative perspective. Conclusions are drawn in section VI.

II. Related works

In our previous work [14] we discussed different approaches to dimensionality reduction methods from the perspective of

[1] github.com/DinarZayahov/thesaurus

textual data. We roughly divided the methods into *global* [1], [11] and *local* [5], [7]. Global methods minimize the effect on some global characteristics like variance or distance function approximation accuracy, while local methods tend to care more about local relations. Among promising global methods of dimensionality reduction there is a family of multilinear tensor methods [13], [15], [16]. They are very intuitive and impressive when deal with multiple data modes. Our work addresses two-mode data (features and items), thus in such a degenerate case major tensor methods (Tucker decomposition, PARAFAC, Tensor Train) are equivalent to SVD. We considered SVD in our previous work and preferred non-linear methods. We showed that for text data we prefer to preserve local communities and keep small distances unchanged, thus we proceeded with local methods. We specifically chose t-SNE [5] as it is often mentioned in the context of text embedding visualization. In this section, we also concentrate on two local methods: t-SNE and SOM [7].

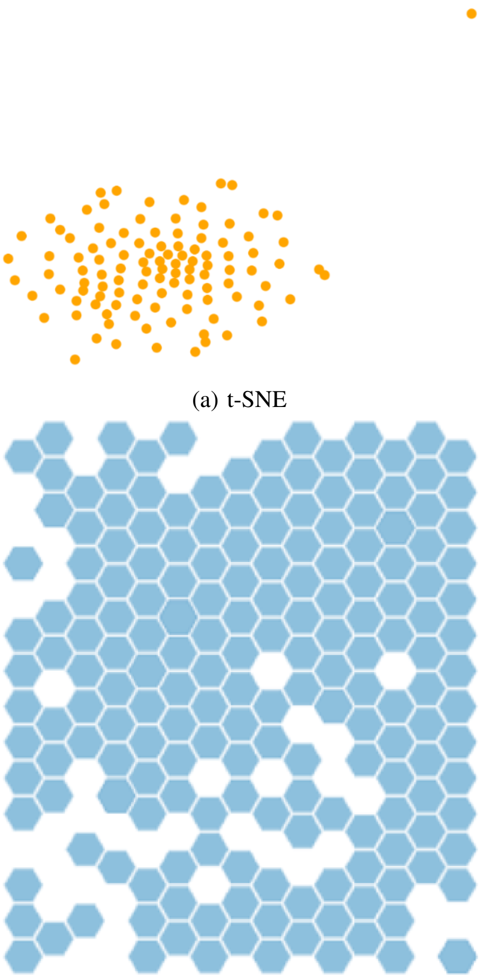
The t-SNE algorithm is mostly tuned by two hyperparameters: perplexity and learning rate. *Perplexity* is related to the number of nearest neighbors considered. Larger data sets usually require a larger perplexity, but this number should be less than the number of datapoints.

If the *learning rate* is too low, most points may look compressed in a dense cloud with few outliers, also low learning rate can lead to non-optimal local minima. The learning rate is related to the number of steps to convergence: changing the learning rate may require changing the number of iterations.

Tuning the t-SNE is a hard task. The algorithms is slow even on thousands of data points. Application of practical optimizations (multi-threading, computational approximations) leads to unstable execution: the same parameters set even with a fixed random seed can converge to a different result.

The self-organizing maps are similar to the t-SNE in terms of idea. The main hyperparameters are also related to neighborhood size and learning rate. In the chosen MiniSom [18] implementation they are referred to as `sigma` and `learning_rate`. The difference comes in the layout. SOM predetermines the form of the target space, and the number of cells that datapoints can occupy. The target space is called `map space` which consists of components called `neurons`. While t-SNE allows points to run away and spread with no fundamental restriction, SOM forces the data to settle inside the rectangular boundaries. This behavior (to our feeling) better corresponds to the task of data visualization on the screen or on the paper. Comparison of two methods is given at Figure 1.

Vector-based methods allow one to obtain latent space embeddings for words, sentences, and texts in an unsupervised manner. Word2vec [9] and similar models do it for particular words and have some issues related to homographs and words out of vocabulary. The attention mechanism [17] and transformer architecture show significant advances in natural language processing solutions. The models of this family are sensitive to a context and morphology by design. Because they target word-in-a-context representations, these models should be used wisely with word-only embeddings.


(a) t-SNE

(b) SOM

Fig. 1: Comparison of t-SNE and SOM on the same data set

Our research is generic with respect to applications, but during our studies we accepted the critical importance of the background language. For our English examples, we chose 32719 words (verbs and nouns) COCA data set [2].

III. Methodology

The very final goal of our research is to deliver a good tool for text data set visualization. We limit ourselves to the following scenario. For any visualization task we introduce a *background set*, which represents the language is general. This can be a generic English language represented by 60K most frequent words, or a specialized language like Russian literature of the XIX century, or quantum physics of the XXI century collected from the preprints. This background set is used for dimensionality reduction model training and can be distributed as a pretrained model together with the embedding tool. And there are also *foreground sets* (can be none or multiple) which are mapped to the background. Visual comparison between foreground and background can highlight missing terms, novel terms, or topic clusters. Comparison

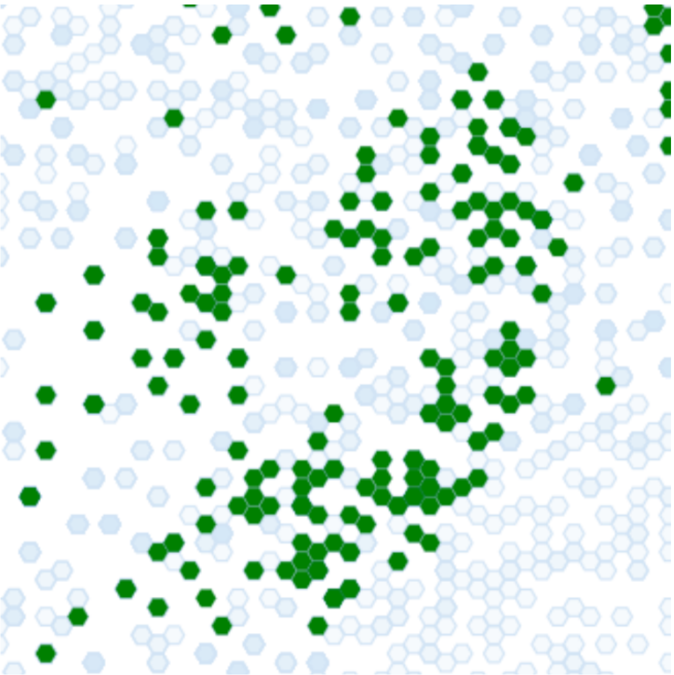


Fig. 2: Example data set visualization. Background set is the hexagonal blue tiles, foreground set is big green hexagons

among the foregrounds can be used for temporal analysis, style matching, or detection of narrow area specific terms. The example of the visualization is present at Figure 2.

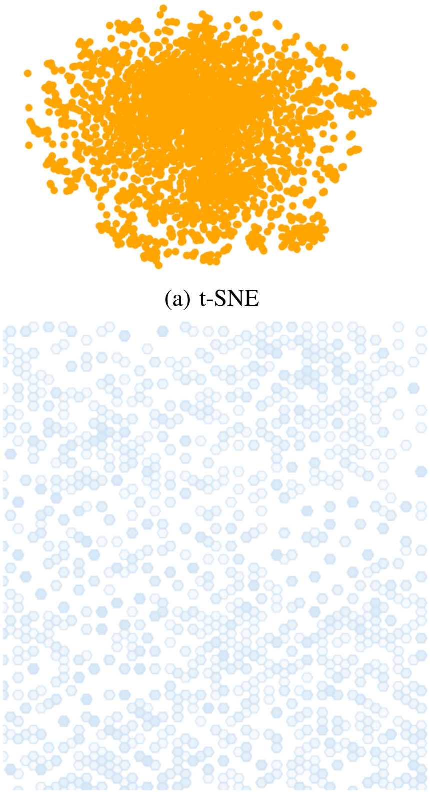
We start our exploration with a comparison of local dimensionality reduction methods. In our previous work [14] we justified this choice compared to global methods. Here, we explore the details of the method. We pay attention to the following factors: *convergence time*, *distribution homogeneity*, *subjective cluster quality*, and *method implementation stability*. We made the comparison using MiniSom [18] implementation of SOM, UMAP [8] implementation of the Uniform Manifold Approximation Projection (UMAP) method and t-SNE [12]. We paid great attention to two things. First, we checked if the method converges to the same state given the same data and hyperparameters. We observed, that t-SNE and UMAP methods even if they accept the random seed, are not stable in terms of the final image. We did not proceed with these methods, as we believe that our results should be reproducible. Second, we visually assessed the resulting images given different original data set sizes. We came up to the conclusion that the sparse results, which we were getting from the above-mentioned methods, did not satisfy our initial purposes that the map should have some bounds for convenience.

We also did a comparison of word embedding models. We considered `en_core_web_sm` and `en_core_web_md` English models from SpaCy [6]. We faced the problem of vectorization of non-dictionary words. For vectors which are zeros in SpaCy model we used averaged embedding of BERT [4] tokens.

As the final stage after we made a choice in favor of SOM, we explored in details how data set size and hyperparameters affect visualization time and quality. Also we used caching for word embeddings as well as for winning neuron positions of words. We present our results in the next section.

IV. Results

In this section we show the results of our tests. Figure 3 shows the cases of t-SNE and SOM applied to 32719 words (nouns and verbs) of the COCA data set. One should pay attention to the shades in SOM image. While t-SNE displays one word per point, SOM can allocate multiple words in one hexagon. Thus, the color intensity corresponds to the local word density.



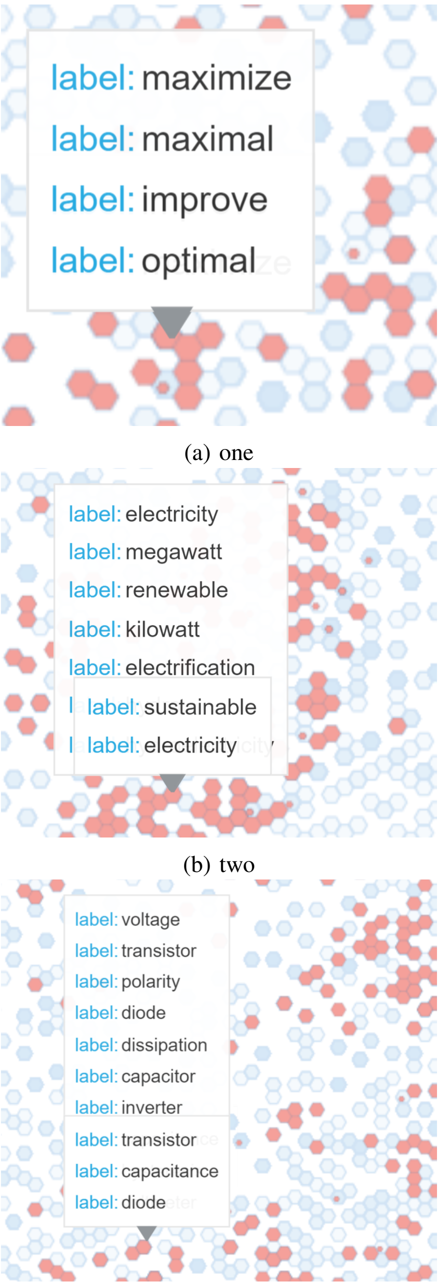
(a) t-SNE

(b) SOM

Fig. 3: COCA data set represented with t-SNE and SOM

To assess the quality of clustering we manually validated words distribution: uniform point distribution does not mean that close words are actually clustered together. After exhaustive grid parameter search we chose *sigma* = *2*, *learning rate* = *5* and *number of iterations* = *50 000* to form a good clustering on our COCA data. Figure 4 shows the quality of clusters in physics texts foreground set.

We also validated that our method can capture a difference between two texts and display them properly. Figure 5 represents a comparison of one set of physical texts against another one. Please, pay attention that there is a large cluster of shared



(a) one

(b) two

(c) three

Fig. 4: Examples of clusters

vocabulary in the center of the map, while edges show specific differences. The reason why we have so many intersections is that both set of texts are from one field and most part of the vocabulary does not change.

We could significantly improve user experience since our previous implementation. To train a new model on a single machine from scratch we need 2 hours for 32719 unique background words even while using more sophisticated embedding model. Table I shows all time characteristics of our method performed on a single CPU Google Colab machine.

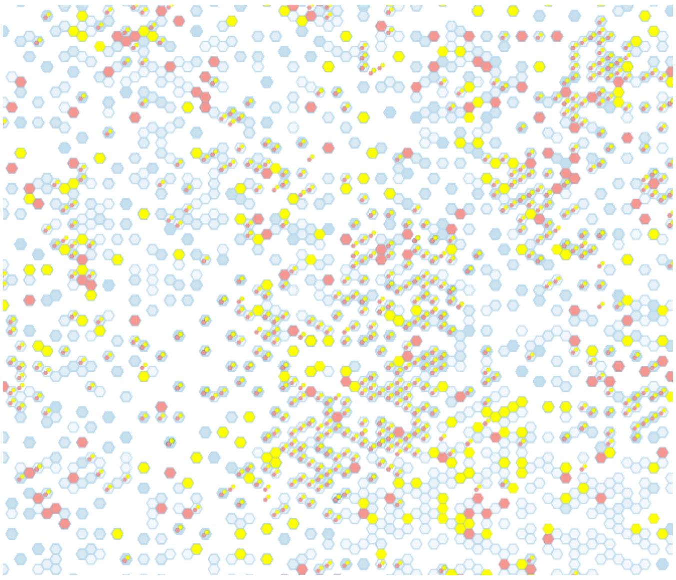


Fig. 5: Two texts: 10 physics articles of year 2010 (red) and 10 articles from year 2020 (yellow) on the common COCA background

Words	Embedding of a foreground set with a pretrained background, min	SOM, min	Visualization, min
1500	0.5	4	0.5
5500	4	17	1.5
11500	30	36	4.5

TABLE I: Time to perform separate stages of the visualization pipeline

V. Discussion

The choice of the dimensionality reduction methods was done in two stages. The first stage was described in our previous paper [14], and referred to the purpose of reducing the dimensions of the textual data. We narrowed our search to local methods and were choosing among t-SNE, UMAP, and SOM. t-SNE was excluded due to unpredictable point distribution, which did to suite screen an paper visualization purposes. UMAP did not show stable behavior even with a fixed random seed parameter. SOM could overcome these difficulties and became our only candidate.

The hard task was to obtain good hyperparameters and derive a set of recommendations for the other background data sets. The exhaustive grid search helped us to find the set of parameters, which is the best in word uniformity and cluster quality. For 32K words of the background set *Sigma = 2*, *learning rate = 5* and *number of iterations = 50000* together with default *activation distance = 'euclidean'* and *neighborhood function = 'gaussian'* gave us satisfactory results for the *hexagonal* topology of the MiniSom. We also observed that with growth of the background set, we need to adjust hyperparameters correspondingly. The increase in the word count in a data set implies the increase of iteration number as well, because the algorithm needs more time to converge. The *learning rate* is a familiar and well-known hyperparameter tuning which will give you either a local or a global optimum state. Next is *sigma* value which determines how data points will be spread on the map, so it should be adequate to the dimensions of the map.

We also observed quadratic construction time growth with the growth of the data set. This can be explained by the specifics of the implementation. We recommend using the pretrained background set and indexes, because the used part of the COCA data contains the most common nouns and verbs and might serve as universal reference for many use cases. But even if user decides to run the algorithm from scratch and train on new corpora, it will be done in a reasonable time. Despite the fact that most of the initial goals were achieved, still there is a place for improvement. For example, although we got logical and meaningful word clusters, the clusters are not highly correlated between each other. And, of course, due to the fact that the set of hyperparameters was picked up by manual selection, the map and distribution can be drawn in a better way.

VI. Conclusion

In this work we aimed to find the best method for displaying text data set on a plain which will capture similarities and clusters, but will not overload the user's perception. We also targeted the creation of software, which can be a handy tool for non-specialists in computer science.

We justified the preference of local dimensionality reduction (t-SNE, SOM) methods compared to global ones (PCA, random projections), as they capture local communities better. We did a comprehensive analysis of two local methods, t-SNE and self-organizing Kohonen maps, highlighted their difference and chose SOM to continue our experiments and implementation.

In this experiments we were able to significantly reduce map construction time compared to our previous work [14]. Now, for the 32719 words background data set the whole fitting and visualization cycle takes 2 hours on a single CPU Google Colab machine. For those users who reuse pretrained background vocabularies visualization is ready in 30 seconds. With these numbers, we are confident that our tool is useful for research groups working with texts.

We want to highlight that a lot of work remains. As English and Russian languages that we considered in our research are well equipped with NLP tools, we could easily implement tokenization, lemmatization, and embedding. This can be different for less common languages. We want to pay additional effort for multilingual support in our future research.

Our results, demos, and implementation are available in the open source repository github.com/DinarZayahov/thesaurus.

References

[1] Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 245–250 (2001)

[2] Davies, M.: The corpus of contemporary american english as the first reliable monitor corpus of english. Literary and linguistic computing **25**(4), 447–464 (2010)

[3] Deleuze, G., Guattari, F., Ricke, G.: Tausend Plateaus: Kapitalismus und Schizophrenie. Deleuze, Gilles: Kapitalismus und Schizophrenie. Merve-Verlag (1992). URL https://books.google.ru/books?id=ygLdzgEACAAJ

[4] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018). URL http://arxiv.org/abs/1810.04805. Cite arxiv:1810.04805Comment: 13 pages

[5] Hinton, G.E., Roweis, S.: Stochastic neighbor embedding. Advances in neural information processing systems **15** (2002)

[6] Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). DOI 10.5281/zenodo.1212303

[7] Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological cybernetics **43**(1), 59–69 (1982)

[8] McInnes, L., Healy, J., Saul, N., Grossberger, L.: Umap: Uniform manifold approximation and projection. The Journal of Open Source Software **3**(29), 861 (2018)

[9] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

[10] Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)

[11] Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science **2**(11), 559–572 (1901)

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

[13] Pham, T.T.: Visualization and classification of neurological status with tensor decomposition and machine learning (2019)

[14] Protasov, S.: An approach to visual thesaurus exploration: a case study for russian language. In: Proceedings of XXI international conference. Informatics: problems, methods, technologies, pp. 1335–1342. Voronezh (2021)

[15] Renato Pajarola, R.B.R.: Tutorial: Tensor decomposition methods in visual computing. https://www.ifi.uzh.ch/dam/jcr:ded4873d-64d8-4ecf-b60f-2fb2742d9c16/TA$_{T}utorial_{P}art1.pdf$

[16] Sozykin, K., Chertkov, A., Schutski, R., Phan, A.H., Cichocki, A., Oseledets, I.: Ttopt: A maximum volume quantized tensor train-based optimization and its application to reinforcement learning (2022). DOI 10.48550/ARXIV.2205.00293. URL https://arxiv.org/abs/2205.00293

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[18] Vettigli, G.: Minisom: minimalistic and numpy-based implementation of the self organizing map (2018). URL https://github.com/JustGlowing/minisom/