Assessment of Document Similarity Visualisation Methods

Mateusz Gniewkowski and Tomasz Walkowiak(✉)

Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{mateusz.gniewkowski,tomasz.walkowiak}@pwr.edu.pl

Abstract. The article deals with the problem of assessing a visualization of the similarity of documents. A well-known approach for showing the similarity of text documents is a scatter plot generated by projecting text documents into a multidimensional feature space and then reducing the dimensionality to two. The problem stems from the fact that there is a large set of possible document vectorization methods, dimensionality reduction methods and their hyperparameters. Therefore, one can generate many possible charts. To enable a qualitative comparison of different scatter plots, the authors propose a set of metrics that assume that the documents are labeled. Proposed measures quantify how the similarity/dissimilarity of original text documents (described by labels) is maintained within a low-dimensional space. The authors verify the proposed metrics on three corpora, seven different vectorization methods, and three reduction algorithms (PCA, t-SNE, UMAP) with many values of their hyperparameters. The results suggest that t-SNE and fastText trained on the KGR10 dataset is the best solution for visualizing the semantic similarity of text documents in Polish.

Keywords: Similarity visualization · Dimensionality reduction · Document embedding · NLP

1 Introduction

Large text corpora are the basic resource for many researchers in humanities and social science [20]. Therefore, there is a need to automatically categorize documents in terms of subject areas. One solution to this problem is to apply supervised text classification methods. The results reported in the literature [25, 28] are very promising, especially those based on BERT [3] deep neural networks. They show that it is possible to automatically assign text documents to subject categories. However, supervised approaches are very often hard to be applied in real-world scenarios, because in practice, most of the analysed by researcher corpora are lacking a consistent set of labels. Developing such labels is a costly process that also requires annotation rules. One could use an already trained classifier to process a new dataset, but it is highly probable that the documents that had been used for training concerned other areas. Supervised models work

Z. Vetulani et al. (Eds.): LTC 2019, LNAI 13212, pp. 348–363, 2022.
https://doi.org/10.1007/978-3-031-05328-3_23

well only used on texts similar to the training data. Therefore, unsupervised approaches like clustering [4,11,29] or document similarity visualisation in 2-D space [19,30] are essential in practise.

Clustering and document similarity visualisation are quite similar processes. In most cases, they are based on representing documents by multidimensional feature vectors (document vectorization), calculating the similarities or distances between those vectors, and then applying clustering [6] or dimensionality reduction algorithm [1]. Within this paper we will focus on the second problem.

The goal of similarity visualisation is to present documents (represented by feature vectors) as points on a 2D plane. Documents that are more similar should be closer together in the plot than objects that differs. Such charts allow people to easily interpret the corpus and find potential outliers. Often, one can find nonobvious relationships between groups of texts that exhibit subtle similarities hidden to the naked eye but traceable by multidimensional statistical techniques [20]. Similarity visualisation is also a very helpful tool in the process of defining labels for future supervised learning experiments.

The main problem in the application of similarity visualisation methods is the selection of the method parameters. First, there are a large number of possible techniques of representing documents by feature vectors, starting from the bag-of-words technique [26], thorough word embedding [9], to deep neural network models like ELMo [18] or BERT [3]. Next, there are many available pretrained language models. There are also many dimensionality reduction algorithms like PCA [6], t-SNE [10], or UMAP [12], and each of them has many hyperparameters. It raises the question of which combination of the above should be selected for visualization? There is no easy answer to that, because the result of the similarity visualisation is a scatter plot (see Fig. 1) that is interpreted by people. The aim of the paper is to find metrics that are consistent with human perception and allow to automatically compare different approaches of generating plots. Such assessments will not only make it possible to generate better visualizations but also will allow easier selection of any parameter (like the vectorization method) for the dataset that is yet to be labeled.

This work is an extension of the research presented in [30]. We added a new corpus, used new methods of generating document vectors, applied new methods of dimension reduction (originally we only considered t-SNE), and finally proposed a much larger set of evaluation metrics (originally 1, and now 5). The metrics, vectorization and reduction methods were evaluated on labelled (in terms of subject area) corpora in Polish.

The paper is structured as follows. In Sect. 2 we shortly describe the vectorization methods that we used to transform documents into feature space. Next, in Sect. 3 we describe the three dimensionality reduction methods that were used in experiments. Section 4 contains descriptions of the proposed evaluation metrics. In Sect. 5 we discuss the datasets and our results. Conclusions are at the end of the paper.

2 Vectorization Methods

2.1 TF-IDF

The TF-IDF method is based on the bag-of word concept [22], i.e., counting the occurrences of the most common terms (words or their n-grams) in the corpus (term frequencies). Next, these frequencies are weighted by the maximum term frequency in a document and by the inverse document frequency. In the performed experiments, we have used the 1000 most frequent terms (words or bigrams).

2.2 fastText

The big step in the area of text analysis was the introduction of the word2vec method [9]. In this approach, individual words are represented by high-dimensional feature vectors (word embeddings) trained on a large text corpus. The most common solution to generate the document features is to average vector representations of individual words. This approach is known as doc2vec [13].

Due to a large number of word forms in morphological rich languages such as Polish, there are two main approaches: to use lemmas (the text have to lemmatized) or the word2vec extension [5] from the fastText package. The last one uses the position weights and subword information (character n-grams) that allow to generate embeddings for unseen words.

Doc2vec as well as TF-IDF ignores word order. Therefore, these methods are not aware of word contexts.

2.3 ELMo

The newest approaches in language modeling are inspired by deep learning algorithms and context-aware methods. The first successful is called ELMo [18]. ELMo word embeddings are defined by the internal states of a deep bidirectional LTSM language model (biLSTM), which is trained on a large text corpus. What is important, ELMo looks at the whole sentence before assigning an embedding to each word in it. Therefore, the embeddings are sentence aware and could solve the problem of polysemous words (words with multiple meanings). As the document feature vector, we used the average mean vector of every sentence in it. Generating sentence vectors is built into the model and consists in mean pooling of all contextualized word representations. The main problem with ELMo is its slow performance caused by the bidirectional architecture of LSTM networks.

2.4 BERT

The next step was a usage of the transformer [27] architecture for building language models. The state-of–the–the-art solution is BERT [3]. Due to its bidirectional representation, jointly built on both the left and the right context, BERT looks at the whole sentence before assigning an embedding to each word in it. As the document feature vector, we have used the CLS pooling method, i.e., the embedding of the initial CLS token.

2.5 Method Summary

The above vectorization methods, except TF-IDF, require pretrained language models. The names used in results reporting and sources of the used models are presented in Table 1.

The main drawback of ELMo and partly BERT is the requirement of using GPU even for the vector generation phase. Usage of ELMo on CPU is impractical due too very long processing time. It is slightly better in the case of BERT, but still TF-IDF and doc2vec work much faster on CPU than BERT.

Table 1. Document vectorization methods and sources of language models

Name	Method	Address
kgr10	fastText	hdl.handle.net/11321/606
kgr10-lemma	fastText	hhdl.handle.net/11321/606
fasttext	fastText	https://fasttext.cc/docs/en/crawl-vectors.html
elmo	ELMo	vectors.nlpl.eu/repository/11/167.zip
tfidf	TF-IDF	–
herbert-kgr10	BERT	https://clarin-pl.eu/dspace/handle/11321/851
herbert-base	BERT	https://huggingface.co/allegro/herbert-base-cased

3 Reduction Methods

The aim of the reduction is to present documents in the 2D plane to visualise the distances or dissimilarities between them. The distances between points should reflect similarities in the original multidimensional space of feature vectors (generated as described in Sect. 2).

There are several methods that can be used for 2D visualisation of multidimensional feature vectors. They can be divided in two categories [12]: ones preserving the distance structure within the data such as the PCA [6] or multidimensional scaling [1] and ones that preserve the local distances over the global distance like t-SNE [10], Laplacia eigenmaps, Isomap, and the newest UMAP [12]. Within this work have analysed three methods: PCA, t-SNE, and UMAP.

3.1 PCA

PCA (Principal component analysis) [17] is a traditional and widely used dimensionality reduction technique. It works by identifying the linear correlations with preserving most of the valuable information. PCA algorithm is based on the principal components of the covariance matrix – a set of vectors, the first of which best fits (explain the maximum amount of variance) the data while the rest are being orthogonal to it. To generate low dimensional space, we ignore the less significant principle components by projecting each data point.

3.2 T-SNE

T-SNE, proposed in [10], is a non-linear dimensionality reduction method. It preserves the similarity between points defined as normalised Gaussians. Therefore, it uses Euclidean distance in the original space. The bandwidth of the Gaussian is set by the bisection algorithm, in a way that the resulting perplexity is equal to some predefined value. As a result, the bandwidth, and therefore the similarity, for each point is adapted to the local density of the data. The similarities in low-dimensional space are modeled by a normalised t-Student distribution. The t-SNE method minimises the Kullback-Leibler divergence between the similarities in both spaces with respect to the locations of the points in the low-dimensional space.

3.3 UMAP

Uniform manifold approximation and projection (UMAP) [12] constructs a high-dimensional graph representation of the data and next optimizes a low-dimensional graph to be as structurally similar as possible. It assumes that the data is uniformly distributed on Riemannian manifold which is locally connected [12]. UMAP high-dimensional graph edges represent the likelihood of connection of each pair of data points. UMAP connects (edges) only points for which the local point radius overlaps. Each point local radius is set based on the distance to each point's and number of neighbours. The size of point neighbours is the method hyperparameter. In many papers, it was shown that UMAP outperforms other methods (including PCA and t-SNE) [12,16,24].

4 Evaluation Methods

The main problem addressed in the paper is the measurement of the document visualization quality. Corpus of text is mapped to the multidimensional space by one of the methods described in Sect. 2. Next, this set of high-dimensional vectors (each representing a single document) is projected to a 2D space using one of the methods described in Sect. 3. As a result, we obtain plots like those presented in Fig. 1. The question is: which combination of document feature vector generation methods, reduction methods, and their hyperparameters should be used? Or, in other words, how to quantify individual plots to be able to choose the best one. We need a metric that allows to compare visualisation results automatically, a metric that promotes results with well-separated classes.

This problem does not have a common quality metric. Therefore, in this section, we propose five different methods. All these metrics are based on the assumption that for method comparison purposes we have a set of labels assigned to the documents. We assume that the documents within the same label are similar (at least some of them, a group does not have to be unimodal) and documents assigned to two different labels are different. In other words, the points representing the same label documents should be placed in low-dimensional space close to each other and far away from points representing other classes.

4.1 Closest Match (CM)

In [30], we proposed a simple coherence score defined as an average (over all points) of a number of k-nearest neighbours belonging to the same class, i.e.:

$$\frac{1}{nk}\sum_{p}\sum_{o\in N_k(p)} I(c(p) == c(o)), \tag{1}$$

where n is a number of points (documents), I is the identity function, $N_k(p)$ is the neighbourhood of p defined by the k closest points (using euclidean distance in low dimensional space), and $c(p)$ is a class of point p. The method is parameterized by k - a number of nearest neighbours used in analysis. It measures how many neighbours (in average) of a given point belong to the same label. In our experiments, we used k equaled to 10. In [30] we shown that the value of the metric depends on k in a similar way regardless the used dataset. Therefore, the value of k is not essential (except the extreme values) in the case of comparison.

4.2 KNN

To measure the quality of the reduction, one could also use any classifier that is trained using two-dimensional data. Therefore, we generated ten folds (90% of the data were used for training) using the stratified K-fold strategy and calculated the average accuracy (Exact Match Ratio, MR) as a final score. The formula is as follows:

$$MR = \frac{1}{n}\sum_{i=1}^{n} I(y_i == \hat{y}_i), \tag{2}$$

$$KNN = \frac{1}{K}\sum_{k} MR_k, \tag{3}$$

where K is the number of folds, I is the indicator function, y_i is a true label of a sample i and $\hat{y}_i$ is predicted label of the same sample.

We decided to use a simple KNN classifier (using ten nearest neighbours), which makes the score similar to the one from the previous section. However, almost any classifier could be used here. We have originally started with the multilayer perceptron (MLP) [6]. However, MLP has a much higher computational cost compared to KNN, and within a preliminary experiment gave close to KNN results. Similar approach, i.e., the KNN classifier, was proposed in [12].

4.3 ARI

Instead of using a classification algorithm, it is possible to use any clustering method. If the clusters obtained in a lower space match a ground-truth label,

then the clusters should be visually separated. We use the adjusted rand index [7] to calculate the correspondence.

$$n_{ij} = |X_i \bigcap Y_j|, \quad a_i = \sum_{j=1}^{g} n_{ij}, \quad b_j = \sum_{i=1}^{p} n_{ij}, \tag{4}$$

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}, \tag{5}$$

where $X = \{X_1, X_2, ..., X_g\}$ defines ground truth labels, $Y = \{Y_1, Y_2, ..., Y_p\}$ defines predicted labels (clusters obtained by the algorithm), g defines number of true labels and p of predicted ones (it is a parameter in the clustering algorithm that we established to be equal g). In the performed experiments we use agglomerative clustering [2] with ward linkage.

4.4 Internal Similarity (INT-SIM)

The next metric we propose to use is the mean distance between samples in the same cluster converted to a similarity measure, i.e.:

$$D(X) = \frac{1}{|C_x|} \sum_{i,j \in C_x, x \in X} d(x_i, x_j), \quad T = \frac{1}{|S|} \sum_{k=1} D(S_k), \tag{6}$$

$$\text{INT-SIM} = \frac{T}{T+1}, \tag{7}$$

where d is a distance between two points, C_x is a set of pairs of points in a cluster X and S is a set of clusters defined by labels. The maximum value of this score is obtained when samples from the same label are gathered in a single coordinate, which can be considered as a defect. There is also a problem with clusters that are made up of subgroups that occur in different places (for example, PRESS data in Fig. 1). The score will be lower in this scenario. To overcome this, we propose to use DBSCAN [23] algorithm for each label in the data to obtain subgroups. In the performed experiments the *eps* parameter of DBSCAN was set to the tenth percentile of a distance distribution between samples in the given group.

4.5 External Dissimilarity (EXT-DIS)

And finally, we propose the external dissimilarity score defined as a normalized (divided by the greatest) mean distance between different groups. We calculate the distance between groups as the average distance between samples in one and the other cluster. The final formula is as follows:

$$D(X, Y) = \frac{1}{|X||Y|} \sum_{x_i \in X} \sum_{y_i \in Y} d(x_i, y_i), \tag{8}$$

$$\text{EXT-DIS} = \frac{\overline{D(X,Y)}}{\max_{i,j \in C}(D(X_i, X_j))}, \tag{9}$$

where X and Y are clusters defined by labels, C is a set of pairs of clusters. The score promotes data reduction with similar distances between groups and might lead to solutions with points from the same group concentrated in one place (similarly to the External Dissimilarity).

5 Experiments

5.1 Datasets

In our experiments, we used three collections of text documents in Polish: $Wiki$, $Press$, and $Qual$. All were labelled in terms of subject area, therefore we can assume that the similarity analysed by the metrics introduced in Sect. 4 is semantic one.

The $Wiki$ corpus consists of articles extracted from the Polish language Wikipedia. It was created by merging two publicly available collections [14] and [15]. The original corpus is labeled by 34 subject categories. For clarity of the presented pictures, we have selected a subset of 10 labels, namely: computers, music, aircraft, games, football, cars, chess, coins, shipping, and animation. The resulting corpus consists of $2,959$ elements.

The second corpus, $Press$ [31] consists of Polish press news. There are $6,564$ documents in total in this corpus. The texts were assigned by the press agency to 5 subject categories (diplomacy, sport, disasters, economy, business, and transportation). All subject groups are very well separated and each group contains a reasonably large number of members (ca. $1,300$ documents per label) without big differences among label sizes.

The last data set, $Qaul$ [11] includes documents containing descriptions of qualifications from a Polish public register of the Integrated Qualifications System and descriptions of degrees from Polish universities. The descriptions mainly consist of so-called learning outcomes statements, which characterize the knowledge, skills, and attitudes required to obtain a given qualification or degree. The data were manually labeled. The labels denote the sectors to which the qualifications belong. Similarly to $WIKI$ corpus, we have selected a subset of seven labels, namely: economy, biology, industry, electronics, music, machines, and architecture. The final corpus consists of $1,419$ documents.

5.2 Results

For every corpus and the previously described vectorization and reduction methods (and many different hyperparameters of the last ones), we generated a chart. In total, we obtained almost 3 500 two-dimensional scatter plots and evaluated them using our metrics. To measure the quality and effectiveness of them, we

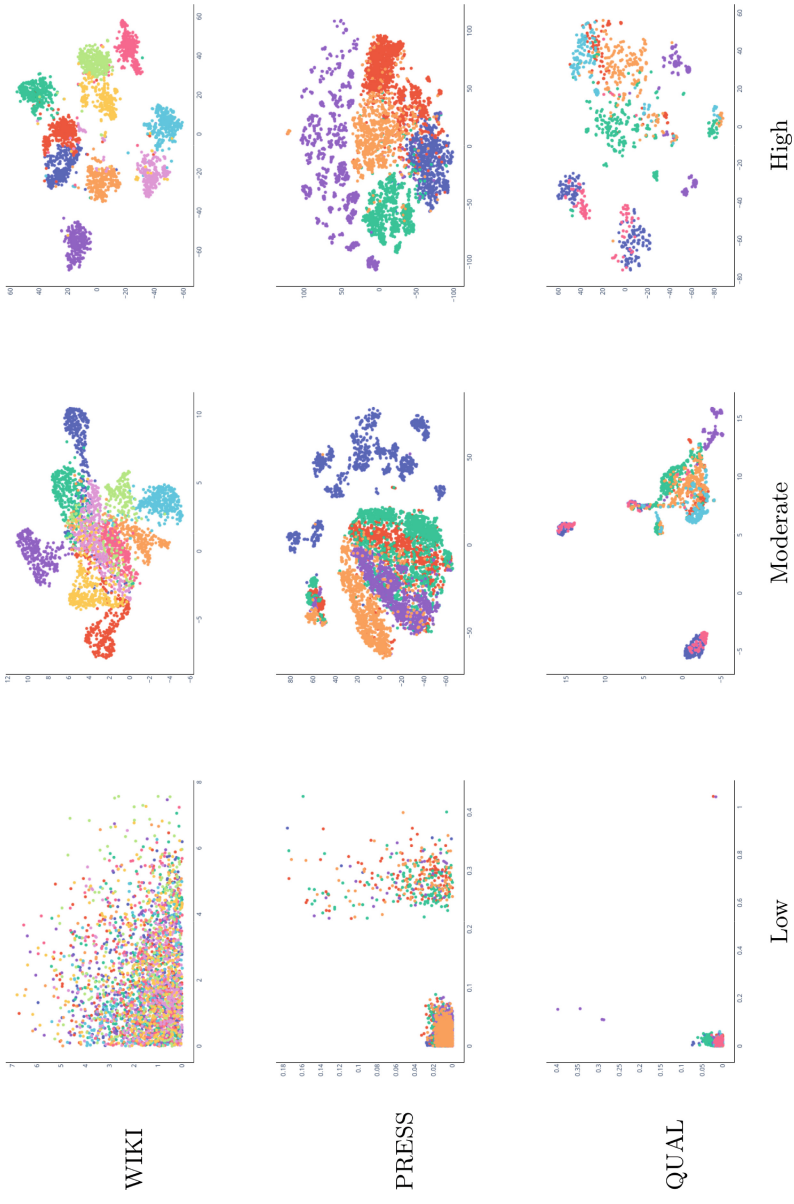

Fig. 1. Exemplary plots related to the highest, lowest, and middle (median) KNN score for every dataset.

Table 2. Scores related to plots in Fig. 1

Dataset	Rating	Scores					Method	Model
		KNN	ARI	CLOSEST MATCH	INT-SIM	EXT-DIS		
WIKI	Low	0.12	0.00	0.20	0.34	0.29	PCA	herbert-base
	Moderate	0.78	0.38	0.76	0.33	0.16	UMAP	tfidf
	High	0.95	0.74	0.94	0.09	0.14	T-SNE	kgr10
PRESS	Low	0.34	0.02	0.36	0.98	0.55	PCA	herbert-base
	Moderate	0.87	0.21	0.85	0.06	0.21	T-SNE	herbert-base
	High	0.94	0.31	0.92	0.03	0.38	T-SNE	kgr10
QUAL	Low	0.37	0.08	0.42	0.97	0.24	PCA	kgr10-lemma
	Moderate	0.76	0.22	0.72	0.55	0.20	UMAP	elmo
	High	0.86	0.32	0.82	0.12	0.16	T-SNE	elmo

Table 3. Statistics of all analysed metrics (scores) for all three corpora.

Method	Score	PRESS			QUAL			WIKI		
		MAX	MIN	AVG	MAX	MIN	AVG	MAX	MIN	AVG
PCA	KNN	0.66	0.34	0.55 ± 0.10	0.61	0.37	0.47 ± 0.08	0.50	0.12	0.29 ± 0.12
	ARI	0.41	0.01	0.14 ± 0.10	0.34	0.05	0.18 ± 0.10	0.25	0.00	0.09 ± 0.08
	CM	0.64	0.36	0.54 ± 0.08	0.59	0.38	0.48 ± 0.06	0.47	0.20	0.31 ± 0.09
	INT-SIM	0.98	0.32	0.85 ± 0.22	0.99	0.43	0.88 ± 0.18	0.95	0.28	0.80 ± 0.23
	EXT-DIS	0.61	0.26	0.43 ± 0.11	0.40	0.13	0.26 ± 0.07	0.50	0.06	0.29 ± 0.13
UMAP	KNN	0.89	0.58	0.75 ± 0.10	0.80	0.53	0.67 ± 0.08	0.88	0.16	0.51 ± 0.22
	ARI	0.39	0.00	0.19 ± 0.08	0.80	0.05	0.31 ± 0.14	0.52	0.01	0.17 ± 0.15
	CM	0.87	0.58	0.73 ± 0.09	0.77	0.52	0.66 ± 0.07	0.85	0.23	0.50 ± 0.20
	INT-SIM	0.55	0.13	0.29 ± 0.08	0.75	0.24	0.44 ± 0.11	0.59	0.15	0.29 ± 0.10
	EXT-DIS	0.71	0.20	0.37 ± 0.08	0.45	0.07	0.22 ± 0.07	0.53	0.10	0.28 ± 0.11
T-SNE	KNN	0.94	0.71	0.88 ± 0.07	0.86	0.62	0.78 ± 0.07	0.95	0.34	0.78 ± 0.20
	ARI	0.59	0.04	0.25 ± 0.10	0.80	0.04	0.34 ± 0.13	0.74	0.01	0.41 ± 0.24
	CM	0.92	0.71	0.86 ± 0.06	0.85	0.61	0.77 ± 0.08	0.94	0.34	0.76 ± 0.20
	INT-SIM	0.14	0.01	0.06 ± 0.02	0.20	0.07	0.13 ± 0.03	0.16	0.00	0.07 ± 0.03
	EXT-DIS	0.62	0.16	0.33 ± 0.10	0.37	0.10	0.20 ± 0.04	0.38	0.07	0.17 ± 0.07

conducted several experiments. The first of them are based on a visual assessment of the correlation between the proposed metrics and the actual plots. In Fig. 1, we present three pictures for each of the corpus that had the lowest, highest, and middle KNN score. In this could be noticed that the top-rated figures contain visually clear and well-separated groups, while the worst-rated ones are rather indistinct. The behavior of KNN measure is following the requirements stated in Sect. 4. Table 2 shows all examined metrics for all plots from Fig. 1. The KNN, ARI, and CLOSEST MATCH (CM) scores act similarly (although the overall promotes different solutions), but the tendency for the remaining scores is the opposite. This means that the best solutions are those where the samples are not too close to each other (INT-SIM) and the distances between pairs of groups are not similar (EXT-DIS). Those two methods should not be used as an out-of-the-box evaluation method.

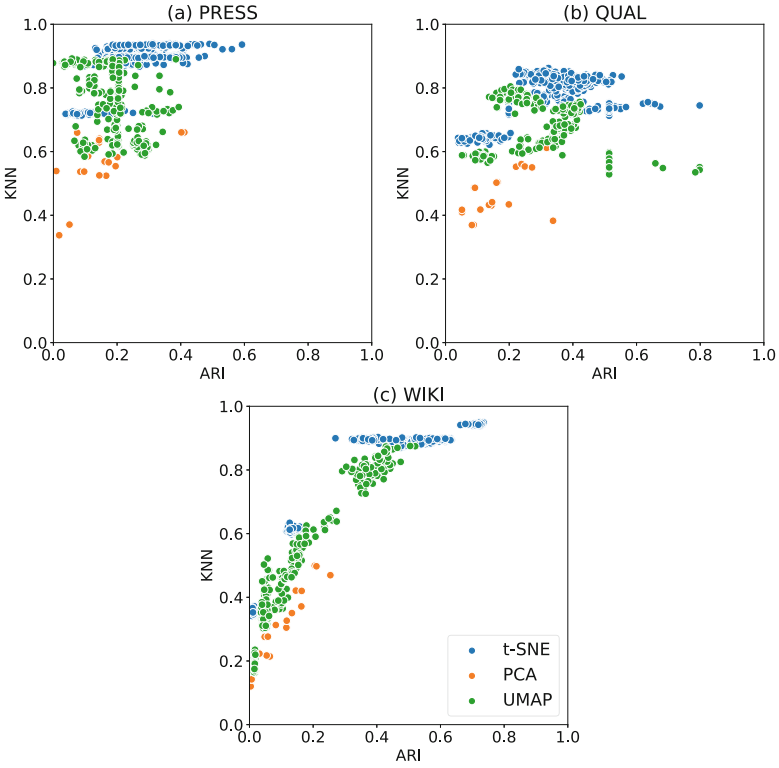

Fig. 2. Values of KNN and ARI metrics (larger values are better) for all methods and data sets. A point represents a single experiment. Experiments differ by data set, document vector generation method and 2-D projection method, and their parameters.

Table 3 shows statistics (average, standard deviation, maximum, and minimum) for each metric and corpus. The statistics are calculated over different metrics, vectorization methods and hyperparameters of reduction algorithms. The Fig. 2 presents the results for each experiment as the relation between KNN and ARI metric for every analysed document in each of the three corpora.

First, it could be noticed that PCA gives the worst results and t-SNE the best. Mind that this statement only applies to the visual aspect of the method. In this paper, we do not address the problem of how well a given method preserves the features of a high-dimensional data. We only focus on the similarity and dissimilarity between documents. Surprisingly, the results for t-SNE outperform UMAP. In the literature, UMAP is considered as a method that outperforms t-SNE [12,24].

Moreover, we can notice that there is a strong correlation between KNN and ARI for *QUAL* dataset, but for the other two corpora the relation disappears. Probably, this is due to the existence of different subgroups within the same label (multimodal data within each label). While KNN takes it into consideration, the ARI score is reduced because it is based on a clustering algorithm with a fixed number of classes (that matches the number of labels in the ground truth). The nature of the used algorithm (i.e., agglomerative clustering) might cause wrong assignments in the low dimensional space. The ARI score should rather be used for uni-modal labels (visually one label point should not occur in different areas) or the number of groups in clustering should be at least twice than the number of labels. Since PCA results are not significant, we focus only on t-SNE and UMAP in further analysis. Figure 3 shows correlation between KNN and ARI metric for left reduction methods. It is not clear to determine which of the vectorization methods works the best. It strongly depends on the corpus (the results follow the intuitive statement that document similarity is subjective) and used a reduction method. However, some tendencies can be noticed. First of all, kgr10 (fastText model trained on the KGR10 corpus) is always in the top three. Secondly, we could notice a high position of a simple and old-fashioned TF-IDF method (red circles), especially for KNN metric. It could be explained by the existence of keywords in each label. For example "aircrafts" (label from *WIKI*) can be simply classified by an occurrence of words such as "aircraft" or "plane". Moreover, the results also suggest that the dataset used for training the language models has a big influence on results. It could be noticed comparing the results achieved by models trained on the KGR10 corpus [8] (pink and violet) to the results obtained by default models (orange and blue respectively). KGR10 results outperforms the base models in the case of fastText and BERT models. Moreover, we see that fastText outperforms (i.e., pink) the Bert based vectorization (violet). The fact that BERT based methods are not suitable for similarity/distance calculation is well known in the literature [21]. The surprising results are for elmo (brown). ELMo have a bad performance for *WIKI* dataset compared to quite good results for *PRESS* and *QUAL*. The other interesting pattern noticeable in the results is the relatively small dependency of the reduction method hyperparameters (i.e., perplexity, learning rate and number of iterations for t-SNE and number of neighbours and minimal distance for UMAP) on the KNN score. The points in the same color represent results from the same vectorization method but with varying values of the reduction method hyperparameters. It could be noticed that they group and even make vertical lines in case of t-SNE. It is probably due to the fact that the perplexity in the case of t-SNE and k-neighbours in the case of UMAP have a big influence on creating subgroups in each label. And as it was already stated, the KNN is less subjective to this feature than ARI.

Comparing the results of t-SNE and UMAP, we can notice that the achieved plots have some similarities, but they differ in details. It shows that each method focuses on different aspects of multidimensional space.

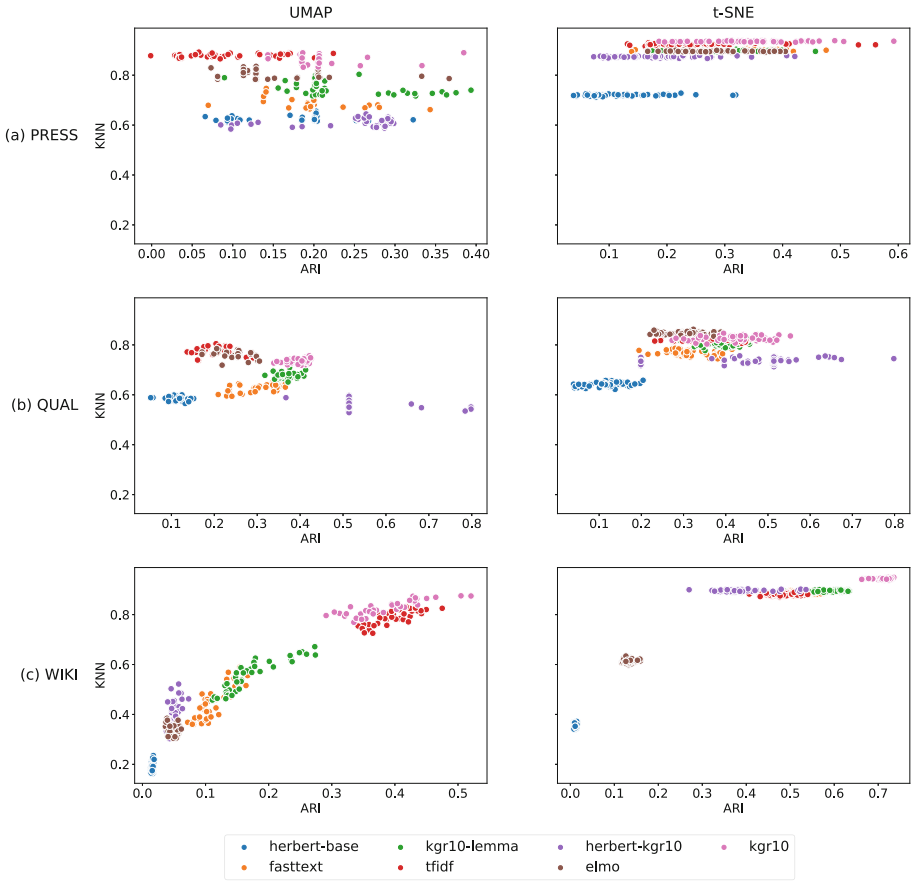

Fig. 3. Values of KNN and ARI metrics (larger values are better) for all three datasets. Points differ by a vector generation method (UMAP and T-SNE) and used hyperparameters.

6 Conclusion

In this work, we proposed a method to assess the visual quality of two-dimensional plots of document similarity obtained using the most popular dimensionality reduction methods. We propose five metrics for quantification of this quality. Based on the testes performed on the three corpora, we conclude that the classifier (KNN), clusterization (ARI) based approaches, or simple score Closest Match can actually determine which of the generated figures better preserves the information of document similarity. This allows for an automatic search of the parameter space to find the optimal ones. We also showed which of the used vectorization methods perform better in the task. The results suggest that fastText based approaches outperform the BERT ones and that the language models for Polish trained on KGR10 outperform others in the analysed problem. Even

though we focused on texts in Polish, our approach can be used in virtually any problem in the field of data mining.

Although we have shown a convenient way to evaluate the appearance of the plots, there are several aspects that require further research. First, although we believe that the correlation between the human perspective and our scores is true, it is necessary to verify this thesis with a larger number of people using a survey. We hope that having plots evaluated by people, we will be able to suggest a combination of proposed scores as a final method (especially INT-SIM and EXT-DIS which cannot be used alone). Next, we focused on solving the problem with the assumption that ground-truth labels are given. This is not always the case in real-world scenarios, but makes just defining the goal difficult. Building such measures would probably require using also context information (other reductions) and not individual plots.

Acknowledgement. Financed by the European Regional Development Fund as a part of the 2014–2020 Smart Growth Operational Programme, CLARIN - Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19.

References

1. Borg, I., Groenen, P.J., Mair, P.: Applied Multidimensional Scaling and Unfolding, 2nd edn. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73471-2
2. Day, W.H.E., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. J. Classif. **1**(1), 7–24 (1984). https://doi.org/10.1007/BF01890115
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Eder, M., Piasecki, M., Walkowiak, T.: An open stylometric system based on multilevel text analysis. Cognit. Stud. Etudes Cognit. **17** (2017). https://doi.org/10.11649/cs.1430
5. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), pp. 3483–3487 (2018)
6. Hastie, T.J., Tibshirani, R.J., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics, Springer, New York (2009). Autres impressions : 2011 (corr.), 2013 (7e corr.)
7. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985). https://doi.org/10.1007/BF01908075
8. Kocoń, J., Gawor, M.: Evaluating KGR10 Polish word embeddings in the recognition of temporal expressions using BiLSTM-CRF. CoRR abs/1904.04055 (2019). http://arxiv.org/abs/1904.04055
9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
10. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008). http://www.jmlr.org/papers/v9/vandermaaten08a.html

11. Marcińczuk, M., Gniewkowski, M., Walkowiak, T., Będkowski, M.: Text document clustering: Wordnet vs. TF-IDF vs. word embeddings. In: Proceedings of the 11th Global Wordnet Conference, pp. 207–214. Global Wordnet Association, University of South Africa (UNISA), January 2021. https://www.aclweb.org/anthology/2021.gwc-1.24
12. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction (2020)
13. Mikolov, T., Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431. Association for Computational Linguistics (2017). http://aclweb.org/anthology/E17-2068
14. Młynarczyk, K., Piasecki, M.: Wiki test - 34 categories (2015). http://hdl.handle.net/11321/217. CLARIN-PL digital repository
15. Młynarczyk, K., Piasecki, M.: Wiki train - 34 categories (2015). http://hdl.handle.net/11321/222. CLARIN-PL digital repository
16. Parra-Hernández, R.M., Posada-Quintero, J.I., Acevedo-Charry, O., Posada-Quintero, H.F.: Uniform manifold approximation and projection for clustering taxa through vocalizations in a neotropical passerine (rough-legged tyrannulet, phyllomyias burmeisteri). Animals **10**(8) (2020). https://doi.org/10.3390/ani10081406. https://www.mdpi.com/2076-2615/10/8/1406
17. Pearson, K.: LIII. on lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. **2**(11), 559–572 (1901)
18. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
19. Piasecki, M., Walkowiak, T., Eder, M.: Open stylometric system webSty: integrated language processing, analysis and visualisation. CMST **24**, 43–58 (2018). https://doi.org/10.12921/cmst.2018.0000007
20. Pol, M., Walkowiak, T., Piasecki, M.: Towards CLARIN-PL LTC digital research platform for: depositing, processing, analyzing and visualizing language data. In: Kabashkin, I., Yatskiv, I., Prentkovskis, O. (eds.) RelStat 2017. LNNS, vol. 36, pp. 485–494. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74454-4_47
21. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3982–3992. Association for Computational Linguistics, November 2019. https://doi.org/10.18653/v1/D19-1410. https://aclanthology.org/D19-1410
22. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
23. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans. Database Syst. **42**(3) (2017). https://doi.org/10.1145/3068335. https://doi.org/10.1145/3068335
24. Smets, T., et al.: Evaluation of distance metrics and spatial autocorrelation in uniform manifold approximation and projection applied to mass spectrometry imaging data. Analyt. Chem. **91** (2019). https://doi.org/10.1021/acs.analchem.8b05827
25. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) Chinese Computational Linguistics, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16

26. Torkkola, K.: Discriminative features for textdocument classification. Formal Pattern Anal. Appl. **6**(4), 301–308 (2004). https://doi.org/10.1007/s10044-003-0196-8
27. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., Luxburg, U.V., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
28. Walkowiak, T.: Subject classification of texts in Polish - from TF-IDF to transformers. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Theory and Engineering of Dependable Computer Systems and Networks, pp. 457–465. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76773-0_44
29. Walkowiak, T., Gniewkowski, M.: Evaluation of vector embedding models in clustering of text documents. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pp. 1304–1311. INCOMA Ltd., Varna, September 2019. https://aclanthology.org/R19-1149
30. Walkowiak, T., Gniewkowski, M.: Visualisation of document similarities based on word embedding models for Polish, pp. 148–151. Wydawnictwo Nauka i Innowacje, Poznań (2019)
31. Walkowiak, T., Malak, P.: Polish texts topic classification evaluation. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pp. 515–522. INSTICC, SciTePress (2018). https://doi.org/10.5220/0006601605150522