Understanding the Impact of Multi-Text Representations on Clustering Task

Rafika Boutalbi, Karima Boutalbi

Understanding the Impact of Multi-Text Representations on Clustering Task

Rafika Boutalbi
Aix-Marseille University
Marseille, France
rafika.boutalbi@lis-lab.fr

Karima Boutalbi
Université Savoie Mont Blanc
Cegedim Business Services
Annecy, France
karima.boutalbi@univ-smb.fr

ABSTRACT

Today we are able to generate a large set of text representations from the simple Bag-of-Word (BOW) to the recent transformer capturing the semantic and the contextual text meaning. However, we know that there is no best text representation. In fact, if we consider the clustering task, a simple BOW can achieve really good results when the text clusters are distinct (eg. Mathematics and Medicine). Thus, in this work, We would like to study the impact of multi-text representation on clustering task. To this end, we proposed a full extensive study on four datasets to understand if the dimensionality reduction highlights a particular interest in text clustering task. Also, a comparison between multi-text representation through feature embeddings using matrix and graphs similarity matrix. Finally, we study the use of the consensus clustering approach to merge muti-text representations, and we compared the explicit consensus through ensemble approaches and implicit consensus using tensor representations that contain all text representations. All these aspects are studied and the obtained results on four datasets show the embedding-based representation and similarity-based representation are complementary and there is no best data representation.

CCS CONCEPTS

• **Unsupervised learning**; • **Clustering** → *Text data*; • **NLP** → Word embedding; • **Representation learning**;

KEYWORDS

Text clustering, Learning representation, Graphs, NLP, Word embedding.

ACM Reference Format:
Rafika Boutalbi and Karima Boutalbi. 2018. Understanding the Impact of Multi-Text Representations on Clustering Task. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Text clustering is an essential tool for various NLP tasks such as document retrieval, sentiment analysis, community detection in social media using users' reviews, etc. Today, there are multiple text representations to represent text data, from the popular Bag-of-word (BOW) to static and contextual embeddings.

Thus, a corpus of n documents (texts) could be represented in two different ways considering a specific embedding b:

- **Embedding representation**: An embedding matrix which is the original text representation with size $n \times m_b$. Given a dataset of n documents, we compute v different data matrices $\mathbf{E}^b$, $b = 1, \dots v$, each one of size $n \times m_b$, where m_b is the number of features of the text representation b.
- **Similarity representation**: Similarity matrix with size $n \times n$ computed using a cosine similarity between each pair of documents. Given a dataset of n documents, we assign a pairwise similarity measure x_{ij}^b to each pair (i, j) of document samples as part of the $\mathbf{X}^b$ similarity matrix of size $n \times n$, as shown in figure 1. To compute the matrix $\mathbf{X}^b$, we use the corresponding representation $\mathbf{E}^b$ of the documents, which is described in the embedding representation. $\mathbf{e}_i^b$ and $\mathbf{e}_j^b$ are the ith and jth row of $\mathbf{E}^b$ respectively.

In the embedding representation, classical clustering algorithms can be applied to each specific embedding b such as K-means, Gaussian Mixture Model (GMM), Spherical Kmeans, etc. However, when we deal with similarity representation, we need to use graph-based clustering methods. In fact, the computed similarity matrices for each embedding b are assimilated to the adjacency matrices, so graph clustering approaches are more suitable in this situation [1]

Also, some recent works, showed that when we deal with multi-text representations, it is better to combine all representations to capture the advantage of each one [4]. For this end, several approaches exist, from simple consensus clustering (ensemble methods), to implicit consensus [5].

In this work, we will study the impact of multi-text representation using embeddings and similarity matrices. We will try to answer the following three questions:

- **Q1**:*Which representation among embeddings or similarity graphs is the best one regarding the clustering task?*
- **Q2**: *Is dimensionality reduction relevant for embeddings and/or similarity graphs regarding the clustering task?*

[1] **Note: There is a major difference between the *text representation* which refers to different text vectorization such as BOW, Bert, etc, and *data representation* studied in this work, namely embedding and similarity representations.**

- **Q3**: *How do ensemble-based approaches using multi-text representation impact the clustering task?*

2 DATA DESCRIPTION

We are using four real-world benchmark datasets coming from text clustering task where the ground-truth partitions are known: **DBLP2**[2], **DBpedia**[3], **Yelp** [4], and **GitHub – AI – Bio** [5] dataset. Each one of these datasets has a multi-level hierarchy. The datasets are described in Table 1. We used five text embedding methods, namely, Bow (Bag-of-word), Skipgram, XLNET[17], and Sentence-Transformers (S-BERT) [15]. The feature size of each dataset is presented in Table 1. We follow the same processing steps applied in [5]. Finally, we evaluate all algorithms using 30 runs and the three metrics: Accuracy (ACC), Purity, and NMI.

Table 1: Description of textual datasets.

		Documents	Clusters	Features				
				Bow	Entity	Skipgram	XLNET	S-BERT
Datasets	**DBLP2**	2223	3	2500	1210	100	120	384
	GitHub-AI-BIO	1528	2	4994	1643			
	Yelp	5000	2	22454	8008			
	DBpedia	11 049	3	67980	24254			

3 FEATURES MATRIX VS. SIMILARITY MATRIX FOR TEXT REPRESENTATIONS

The objective of this section is to discover if there is any advantage of using embedding or similarity representation. To this end, `K – means` [12], `SphericalK – means` (`SK – means`) [6], and `GMM` [7] are applied to embedding representation.

On the other hand, for similarity representation, we will use graph-based approaches, as similarity matrices are assimilated to adjacency graphs. Graphs are a generic way of modeling relations and interactions, represented by edges with a given weight or distance, between entities represented as vertices (or nodes). They can capture complex interactions into a relatively simple framework, as such they have emerged as fundamental conceptual tools in a large set of scientific domains (biology, neurology, sociology, communication, economics, *etc.*). Thus, `Co – clustMod` [2] , `ColutInfo`, and `SPLBM` [1] are applied to similarity representation.

We compare, over all datasets, the NMI metrics for all algorithms using embedding and similarity representations. Table 2 shows the obtained results. Over **DBLP2** and **Github** data, Similarity representation allows us to attain the best performance for all text representation, only for Sentence-Bert representation, where we observe that embedding representation is better. On **DBpedia** and **Yelp** data which are the biggest datasets in our experiments, there is no best representation between Embedding and similarity for all text representation.

[2]https://github.com/boutalbi/TensorClus
[3]https://www.kaggle.com/code/danofer/dbpedia-hierarchical-text-classification-dl
[4]https://github.com/yumeng5/WeSHClass/tree/master/yelp
[5]https://github.com/yuzhimanhua/HiGitClass

Table 2: Evaluation of clustering in terms of NMI. The bold value represents the best results for each text representation. blue cell represents the best performances among Embedding and Similarity, and the bold cell ones are the second-best performances among Embedding and Similarity.

Data	Representations	Embedding matrix			Similarity matrix		
		`GMM`	`K – means`	`SK – means`	`CoclustMod`	`SPLBM`	`CoclustInfo`
DBLP2	BOW	0.464	0.433	0.599	**0.617**	0.568	0.602
	XLNET	0.428	0.378	0.442	0.431	0.444	**0.454**
	Skipgram	0.375	0.368	0.423	0.411	**0.447**	0.446
	Entity	0.398	0.425	0.424	0.404	**0.43**	0.422
	S-Bert	0.777	**0.853**	0.789	0.752	0.612	0.729
GitHub	BOW	0.592	0.581	0.805	0.731	0.664	**0.818**
	XLNET	0.585	0.585	0.83	0.805	0.618	**0.844**
	Skipgram	0.567	0.558	0.516	0.515	**0.618**	0.516
	Entity	0.579	0.567	0.549	**0.635**	0.618	0.514
	S-Bert	0.916	**0.921**	0.919	0.892	0.688	0.918
Yelp	BOW	0.538	0.563	0.625	**0.636**	0.5	0.628
	XLNET	0.503	**0.523**	0.515	0.516	0.514	0.516
	Skipgram	**0.56**	0.54	0.549	0.557	0.5	0.55
	Entity	0.587	0.636	0.634	**0.642**	0.5	0.634
	S-Bert	0.551	0.566	0.579	**0.665**	0.5	0.575
DBPEDIA	BOW	0.442	0.408	0.676	0.677	0.648	**0.691**
	XLNET	0.436	0.509	0.681	**0.689**	0.624	0.672
	Skipgram	0.423	**0.442**	0.419	0.405	0.344	0.424
	Entity	0.436	0.456	0.509	**0.572**	0.344	0.483
	S-Bert	0.865	**0.995**	0.876	0.799	0.684	0.792

4 DIMENSIONALITY REDUCTION AND MULTI-TEXT REPRESENTATION: A GOOD DEAL?

Figures 2 represents the low-dimensional projection of **DBpedia** for Sentence-Bert representation using four dimensionality reduction algorithms applied on the embedding representation namely, Principal Compoment Analysis (`PCA`) [14], Singular Value Decomposition (`SVD`) [9], t-Distributed Stochastic Neighbor Embedding (`TSNE`) [11], and Uniform Manifold Approximation and Projection (`UMAP`) [13], and four dimensionality reduction algorithms for similarity data representation namely, Correspondance Analysis (`CA`) [8], Non-Negative Matrix Factorization (`NMF`) [10], Singular Value Decomposition named `SVDS` to differentiate the one applied to embedding representation, and Spectral Dimensionality reduction (`SPectral`) [3]. We notice that the embedding-based representation seems to be better for representing this dataset. In table 3, the clustering results using `K – means` on the obtained dimensionality reduction results are reported. We confirmed the obtained assumption on the **DBpedia** dataset, where embedding-based representation is better. However, on the other datasets, it is not clear which representation is better between embedding and similarity.

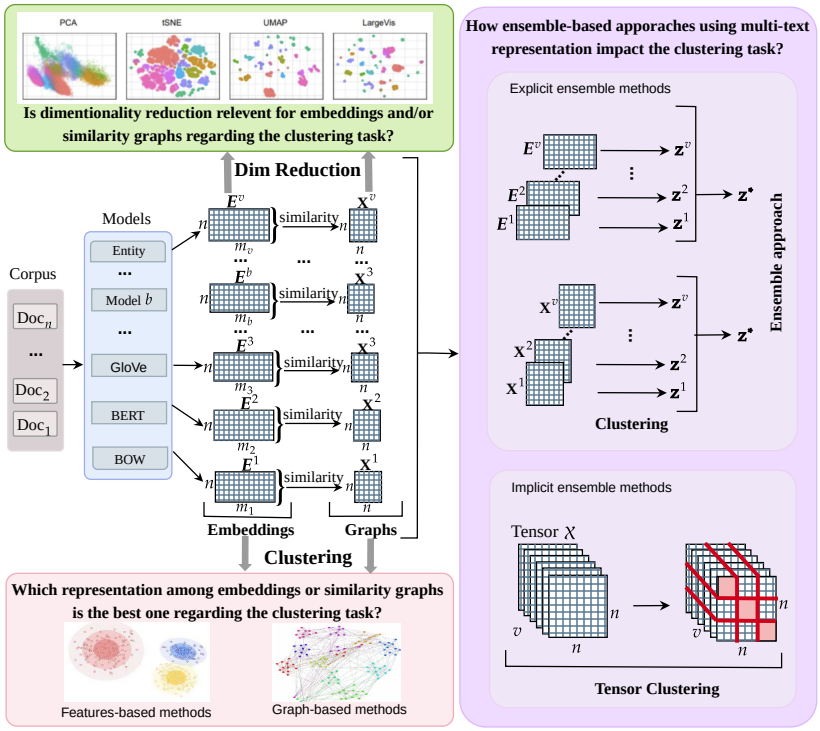

Figure 1: Goal of the proposed methodology.

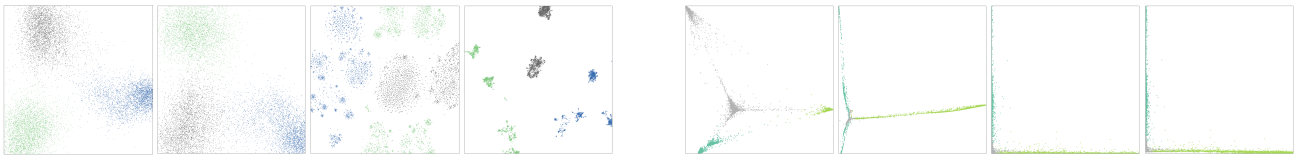

Figure 2: Dimensionaity reduction results on DBpedia dataset.

5 CONSENSUS APPROACH AS A SOLUTION FOR CLUSTERING OF MULTI-TEXT REPRESENTATION

As explained in [5], two consensus clustering approaches exist, the *explicit* and the *implicit* approaches. The explicit consensus clustering uses the clustering results obtained on all text representations and applies the algorithm 1 that uses ClusterEnsembles consensus algorithm `ClusterEnsembles` [16][6],

`ClusterEnsembles` relies on `CSPA`, `HGPA`, and `MCLA` and returns the best results in terms of the mean of NMI between the obtained consensus clustering $\mathbf{Z}^*$ and the different clustering solutions $\{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \ldots, \mathbf{Z}_r\}$.

Using embedding representation, each individual embedding matrix is clustered separately through a single approach, namely `GMM`, `K – means`, and `SK – means`. Using similarity representation, each individual similarity graph matrix is clustered separately through a single graph clustering approach namely, `Co – clustMod`, `ColutInfo`, and `SPLBM`. Thus, a consensus mechanism merges the resulting clusters.

The implicit consensus refers to the joint clustering approach that implements a global clustering over all graphs, by optimizing a global clustering optimization function. For this end, we use a tensor clustering approach named `TGM` [4] which aims to cluster multiple graphs structured on tensor data $\mathbf{X}$, where each slice of the tensor $\mathbf{X}_b$ represent a similarity matrix. `TGM` algorithm optimizes a sum of modularity over all similarity matrices to obtain a unique clustering partition for all graphs (similarity matrices).

We compare the results of the explicit consensus (Algorithm 1) using embedding and similarity representation with the implicit consensus using the joint clustering approach represented by tensor approach `TGM`. The results are shown in Table 4, and suggest that the contribution of multi-graph clustering working as an implicit

[6] https://github.com/827916600/ClusterEnsembles

Table 3: Evaluation of dimensionality reduction in terms of NMI. The bold value represents the best results for each text representation. blue cell represents the best performances among Embedding and Similarity, and the bold cell ones are the second-best performances among Embedding and Similarity.

Data	Representations	Embedding matrix				Similarity matrix			
		T − SNE	PCA	UMAP	SVD	SVDS	NMF	CA	SPectral
Github	BOW	0.629	0.581	0.541	0.581	**0.734**	0.73	0.619	0.586
	XLNET	0.509	0.534	**0.59**	0.529	0.516	0.561	0.571	0.549
	Skipgram	**0.736**	0.549	0.643	0.553	0.664	0.68	0.616	0.672
	Entity	0.579	0.585	0.523	0.585	0.598	0.637	0.618	**0.753**
	S-Bert	0.87	**0.918**	0.91	0.912	0.799	0.835	0.617	0.779
DBLP2	BOW	**0.585**	0.513	0.419	0.535	0.422	0.423	0.43	0.495
	XLNET	0.404	0.398	0.43	0.392	0.435	0.435	0.429	**0.441**
	Skipgram	0.411	0.43	0.43	0.43	0.442	0.385	0.425	**0.465**
	Entity	0.413	0.419	0.42	0.419	0.42	0.42	**0.43**	0.395
	S-Bert	0.67	**0.683**	0.429	0.677	0.549	0.559	0.43	0.549
Yelp − 5K	BOW	0.511	0.564	**0.581**	0.562	0.509	0.504	0.501	0.553
	XLNET	0.532	0.538	0.51	0.538	0.503	0.504	**0.544**	0.509
	Skipgram	0.635	0.634	0.634	**0.638**	0.574	0.525	0.51	0.57
	Entity	0.509	0.519	0.508	0.517	0.509	0.511	0.5	**0.522**
	S-Bert	0.588	0.569	0.533	**0.637**	0.508	0.533	0.501	0.502
DBPEDIA	BOW	**0.589**	0.398	0.472	0.397	0.501	0.501	0.344	0.498
	XLNET	0.348	0.403	0.376	0.403	**0.423**	0.422	0.418	0.382
	Skipgram	**0.628**	0.449	0.422	0.458	0.481	0.476	0.345	0.468
	Entity	**0.635**	0.474	0.495	0.474	0.508	0.509	0.345	0.503
	S-Bert	0.65	0.65	**0.655**	0.649	0.591	0.597	0.516	0.57

Algorithm 1: `Consensus − Algorithm`

Input: $\mathcal{X}$: Various data representation Matrices $1 \ldots v$, `Algo`: Clustering Algorithm , g: Number of clusters

(1) **Initialization:** *AllLabels* =[] ;

(2) **for** $\mathbf{x}_b$ *in* $\mathcal{X}$ **do**

(2.1) Run the clustering algorithm `Algo`$(\mathbf{x}_b, nClusters = g)$;

(2.2) Generate the clustering vector C_b generated by `Algo` ;

(2.3) *AllLabels*.append(C_b);

(3) Run the consensus algorithm `ClusterEnsembles`(*AllLabels*) that generates the consensus clustering vector *ConsenusLabel*;

Return *ConsenusLabel*

consensus is better than the explicit consensus applied as *a posteriori* step in the clustering schema. Nonetheless, `TGM` achieves the best performance in particular for retrieving the clusters on all datasets except for **DBLP2** where `Consensus − Kmeans` is slightly better. Our results match with the results obtained by the `TSPLBM` in [5].

6 CONCLUSION

In this paper, we presented an extensive experimental study to understand the impact of multi-text representations on clustering

Table 4: Comparison of consensus clustering results in terms of ACC, NMI and Purity using Algorithm 1. The bold blue values represent the best performances, and the bold ones are the second-best performances.

Data	Representation	Algorithms	ACC	NMI	Purity
DBLP2	*Embedding*	Consensus − GMM	0.512	0.114	0.554
		Consensus − Kmeans	**0.635**	**0.19**	**0.635**
		Consensus − SKmeans	0.554	0.177	0.579
	Similarity	Consensus − CoclustMod	0.558	0.175	0.581
		Consensus − CoclustInfo	0.546	0.168	0.573
		Consensus − SPLBM	0.552	0.157	0.571
		TGM	**0.591**	**0.185**	**0.591**
Github	*Embedding*	Consensus − GMM	0.704	0.141	0.706
		Consensus − Kmeans	0.709	0.141	0.709
		Consensus − SKmeans	**0.77**	**0.246**	**0.77**
	Similarity	Consensus − CoclustMod	0.763	0.233	0.763
		Consensus − CoclustInfo	**0.77**	0.245	**0.77**
		Consensus − SPLBM	0.579	0.065	0.657
		TGM	**0.859**	**0.402**	**0.859**
Yelp	*Embedding*	Consensus − GMM	0.577	0.018	0.577
		Consensus − Kmeans	0.531	0.003	0.531
		Consensus − SKmeans	0.618	0.041	0.618
	Similarity	Consensus − CoclustMod	**0.641**	**0.058**	**0.641**
		Consensus − CoclustInfo	0.621	0.043	0.621
		Consensus − SPLBM	0.511	0.0	0.511
		TGM	**0.683**	**0.101**	**0.683**
DBPEDIA	*Embedding*	Consensus − GMM	0.663	0.319	0.664
		Consensus − Kmeans	**0.817**	0.49	**0.817**
		Consensus − SKmeans	0.752	**0.498**	0.753
	Similarity	Consensus − CoclustMod	0.733	0.464	0.736
		Consensus − CoclustInfo	0.735	0.485	0.737
		Consensus − SPLBM	0.692	0.425	0.695
		TGM	**0.995**	**0.97**	**0.995**

task. To this end, we used four datasets, several text representation methods, and two data representations namely Embedding and Similarity First, the comparison between the embedding and similarity representation shows a little advantage to similarity representation, even if there is information loss through the data transformation. On the other hand, based on dimensionality reduction approaches, it is not clear which representation is better. Finally, the experimentation using ensemble approaches on multi-text representations showed that implicit consensus using tensor approach `TGM` is highly better than explicit consensus. For future work, we plan to develop a new approach combining the embedding and the similarity representations to take advantage of both representations.

REFERENCES

[1] Ailem, M., Role, F., Nadif, M.: Sparse poisson latent block model for document clustering. IEEE Transactions on Knowledge and Data Engineering **29**(7), 1563–1576 (2017)

[2] Ailem, M., Role, F., Nadif, M.: Co-clustering document-term matrices by direct maximization of graph modularity. In: CIKM. pp. 1807–1810 (2015)

[3] Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.F., Vincent, P., Ouimet, M.: Spectral Dimensionality Reduction, pp. 519–550. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[4] Boutalbi, R., Ait-Saada, M., Iurshina, A., Staab, S., Nadif, M.: Tensor-based graph modularity for text data clustering. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2227–2231 (2022)

[5] Boutalbi, R., Labiod, L., Nadif, M.: Implicit consensus clustering from multiple graphs. Data Mining and Knowledge Discovery **35**(6), 2313–2340 (2021)

[6] Buchta, C., Kober, M., Feinerer, I., Hornik, K.: Spherical k-means clustering. Journal of statistical software **50**(10), 1–22 (2012)

[7] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B **39**, 1–38 (1977)
[8] Dodge, Y.: The Oxford dictionary of statistical terms. OUP Oxford (2003)
[9] Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. In: Handbook for Automatic Computation: Volume II: Linear Algebra, pp. 134–151. Springer (1971)
[10] Lee, D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in neural information processing systems **13** (2000)
[11] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
[12] MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
[13] McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
[14] 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)
[15] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: EMNLP-IJCNLP (2019)
[16] Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research **3**, 583–617 (2002)
[17] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding (2019)