IEcons: A New Consensus Approach Using Multi-Text Representations for Clustering Task

Karima Boutalbi, Rafika Boutalbi, Hervé Verjus, Kave Salamatian, David Telisson, Olivier Le Van

IEcons: A New Consensus Approach Using Multi-Text Representations for Clustering Task

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

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

Hervé Verjus
Université Savoie Mont Blanc
Annecy, France
herve.verjus@univ-smb.fr

Kave Salamatian
Université Savoie Mont Blanc
Annecy, France
kave.salamatian@univ-smb.fr

David Telisson
Université Savoie Mont Blanc
Annecy, France
david.telisson@univ-smb.fr

Olivier Le Van
Cegedim Business Services
Lyon, France
Olivier.levan@cegedim.com

CCS CONCEPTS

• **Unsupervised learning**; • **Clustering** → *Consensus Clustering*; • **NLP** → Word embedding; • **Representation learning** → Tensor;

KEYWORDS

Clustering, Embeddings, Consensus, Implicit consensus, Tensor data

ACM Reference Format:
Karima Boutalbi, Rafika Boutalbi, Hervé Verjus, Kave Salamatian, David Telisson, and Olivier Le Van. 2024. IEcons: A New Consensus Approach Using Multi-Text Representations for Clustering Task. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Text clustering is an important task in NLP for many applications, such as topic modeling, entity linking, question answering, etc. To perform text clustering,

two main types of text representation exist, namely frequency-based and prediction-based methods. Frequency-based methods are statistical methods based on terms frequency of the text, like Bag-of-Word (BOW), TF-IDF, and Skip-gram. These approaches can achieve good results when the text clusters are distinct. In fact, if we consider the clustering task, a simple BOW can obtain a high clustering performance with two very different classes of domain eg. Mathematics and Medicine.

On the other hand, prediction-based methods are based on learning models to predict dense numerical vectors representing texts named *Word Embeddings*. These approaches are able to extract the semantics of a given word or a piece of text, using a fixed-size vector. Word embeddings are classified into two categories (i) static embeddings that are generated using a fixed corpus of documents, thus each word has a unique representation, such as Glove[12], Word2vec [11], etc (ii) and contextual embeddings that consider the right and left context of a given word to generate the vector embedding, such as Bert[9], XLNET [15], etc.

Therefore, besides the challenge of selecting the best text representation in the unsupervised context of text clustering, data representation is also an essential research question in text clustering. 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 (see Figure 1) :

- Feature representation, which consists of using the original feature/text representation (or embedding) $\mathbf{E}^b \in \mathcal{R}^{n \times m_b}$, for a given dataset with n documents and v representations, and m_b is the text representation size of the bth representation.
- Similarity representation, which consists of computing a cosine distance between each pair of text representations. Given a dataset with n documents, a similarity matrix $\mathbf{X}^b \in \mathcal{R}^{n \times n}$ is computed for each text representation $b = 1...v$. Thus, the cell x_{ij}^b of the similarity matrix $\mathbf{X}^b$ contains a pair-wise similarity measure between the two vectors $\mathbf{E}_i^b$ and $\mathbf{E}_j^b$.

However, in the unsupervised context of text clustering, it is challenging to choose/decide which is the best text representation and/or data representation for the clustering task. Figure 2 represents the obtained clustering results on three textual datasets namely DBpedia, Yelp, and Classic3 using five different text representations namely BOW, Skipgram, Entity, XLNET[15], and S-Bert (Sentence-Bert)[13]), and two data representations namely feature representation (or embedding) and similarity representation. We can only apply classical clustering algorithms on the feature matrix representation, such as `Kmeans`[10], `GMM`[8], `Spherical Kmeans` (`SKmeans`) [7], etc. Nevertheless, similarity representation can be assimilated into adjacency matrices or graphs. Therefore, graph clustering methods are used in this case, such as `CoclustMod`[2], `SPLBM`[3], `Louvain`, etc.

Based on the results in figure 2, we observe that there is no best text representation among the five representations, even if S-Bert achieves good results in terms of text clustering in several cases. Also, comparing Embedding and similarity representation shows that no best data representation improves the text clustering task. Based on those conclusions, we need novel approaches that combine different text representations and that consider the two data representations namely, embedding and similarity.

CIKM '24, October 21–25, 2024, Boise, ID, USA.

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/XXXXXX.XXXXXX

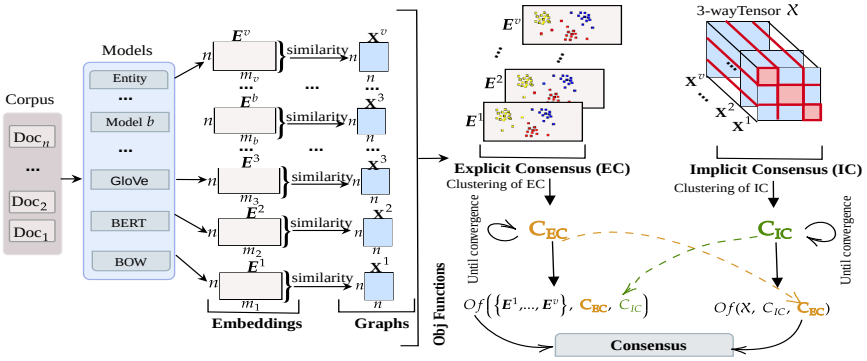

Figure 1: Goal of the proposed Hierarchical Tensor Graph Modularity (HTGM) approach.

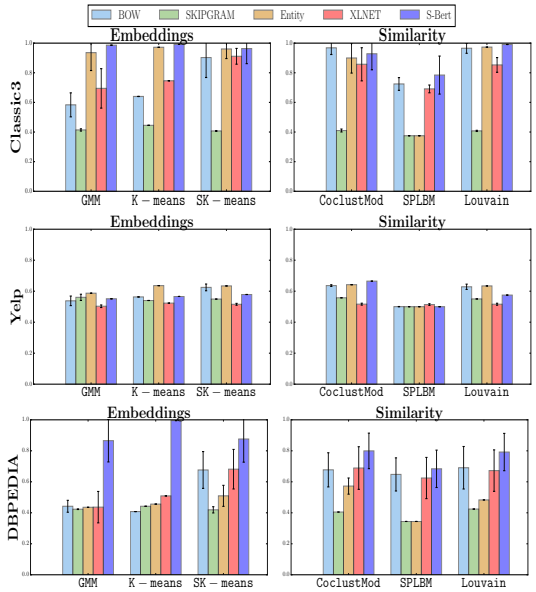

Figure 2: Text clustering performance in terms of NMI using different data and text representations for three datasets.

To be able to consider multi-text representation there are two ways, namely explicit consensus (EC) [14] and implicit consensus (IC) [4, 6]. The explicit consensus can be used on both embedding and similarity representation. In this case, clustering is applied to the embedding matrices or similarity matrices, then the consensus function is applied on the obtained clustering vectors to compute the consensus clustering vector. The implicit consensus is applied to a tensorial structure (or a 3-way tensor) of multiple similarity matrices (see Figure 1). In this case, we use a tensor clustering algorithm that optimizes the clustering simultaneously over all similarity matrices to obtain a unique partition over the matrices representing the implicit consensus clustering vector.

In this paper, we propose a novel algorithm IEcon for multi-text representations – with an unlimited number v of text representations – that combines explicit consensus clustering (EC) and implicit consensus clustering (IC). Figure 1 presents the objective of the proposed clustering algorithm for text data. We evaluated the IEcon algorithm over five real-world datasets and we compared the results with state-of-the-art consensus approaches.

2 PROPOSED METHOD

As highlighted in the introduction each text representation contains some specific information that does not necessarily exist in the other ones. Thus, the consensus is an essential task when we deal with text clustering, allowing to combine several text representations and ensure obtaining the best trade-off [6].

2.1 Explicit Consensus (EC)

The explicit consensus (EC) consists of combining clustering partitions $C_1, C_2, ..., C_v$ obtained from the clustering algorithm applied on all text representations $E_1, E_2, ..., E_v$, respectively. It allows computing a global consensus partition that takes into account the mutual information along clustering partitions (see figure 3). We can cite ClusterEnsembles, which is an ensemble method for clustering proposed by [14] allowing us to compute the consensus clustering vector.

2.2 Implicit Consensus (IC)

On the other hand, the implicit consensus (IC) consists of optimizing a tensor clustering objective function, where the tensor $\mathcal{X}$ contains the similarity matrices computed for all text representation $X_1, X_2, ..., X_v$ – as explained in the introduction section – allowing to capture the mutual information along corpus entries (or texts). Some tensor clustering algorithms can be used for implicit consensus such as [4, 6]. The TGM[1] algorithm presented in [4] is a recent and effective approach that optimizes graph modularity to estimate the implicit clustering partition C for all similarity matrices structured as a 3-way tensor and considering g clusters (see equation 1). The authors showed that the objective function Q evolution of TGM is strongly correlated to mutual information which supports our intuition.

$$Q(\mathcal{X}, Z) = \sum_{b=1}^{v} \frac{1}{x_{..}^{b}} \sum_{i,j=1}^{n} \sum_{k=1}^{g} (x_{ij}^{b} - \frac{x_{i.}^{b} x_{.j}^{b}}{x_{..}^{b}}) c_{ik} c_{jk}. \quad (1)$$

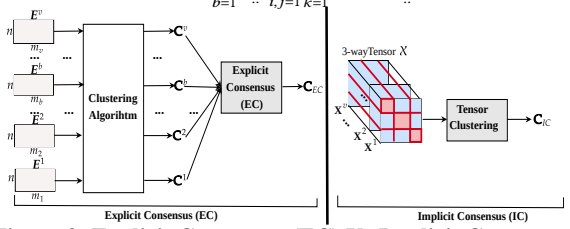

Figure 3: Explicit Consensus (EC) Vs Implicit Consensus (IC).

[1] https://github.com/TGMclustering/TGMclustering

2.3 IEcons: A New Consensus approach combining Implicit and Explicit Consensus

The proposed method IEcons is based on explicit and implicit consensus. The explicit consensus is applied to the embedding representation allowing IEcons to consider the global mutual information from the clustering of all text representations. On the other hand, an implicit consensus clustering using the tensor clustering approach TGM is applied to similarity representations. TGM algorithm is initialized using the partition obtained by ESK (Ensemble Spherical Kmeans). Algorithm 1 shows the different steps of the proposed IEcon algorithm that alternates the implicit and explicit consensus until convergence (considering the normalized objective functions Of_{EC} and Of_{IC} of ESK and TGM respectively) and improves each of them by providing TGM with ESK resulting clustering partition and providing ESK with the TGM resulting clustering partition. A consensus clustering vector is generated to return the final clustering partition. Thus, the IEcons algorithm ensures the robustness of the implicit and explicit consensus at each iteration.

Algorithm 1: IEcons

Input: $\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_v$ List of embedding representations,
$\mathcal{X}$: Tensor of similarity represensentions, g: Cluster number.
(1) **Initialization:** Random initialization of $\text{ESK}(\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_v)$ algorithm at $t = 0$ and obtain the explicit consensus vector $\mathbf{C}^0_{EC}$
(2) Initialize tensor clustering algorithm $\text{TGM}(\mathcal{X})$ at $t = 0$ using $\mathbf{C}^0_{EC}$ and obtain the implicit consensus vector $\mathbf{C}^0_{IC}$
(3) Compute the consensus clustering C^t_{IEcons} using $\text{ClusterEnsembles}(\mathbf{C}^0_{EC}, \mathbf{C}^0_{IC})$
(4) **repeat**
  (4.1) Run $\text{ESK}(\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_v)$ algorithm using as initialization C^t_{IEcons} and obtain $\mathbf{C}^{t+1}_{EC}$
  (4.2) Run $\text{TGM}(\mathcal{X})$ algorithm using as initialization C^t_{IEcons} and obtain $\mathbf{C}^{t+1}_{IC}$
  (4.3) Compute the consensus clustering C^{t+1}_{IEcons} using $\text{ClusterEnsembles}(\mathbf{C}^{t+1}_{EC}, \mathbf{C}^{t+1}_{IC})$
until *Convergence* $(Of^{t+1}_{EC} + Of^{t+1}_{IC}) - (Of^{t}_{EC} + Of^{t}_{IC}) < \epsilon$;
Return $C^{t+1}_{IEcons}, Of^{t+1}_{EC}, Of^{t+1}_{IC}$

3 EXPERIMENTS

3.1 Dataset description

We evaluated IEcons based on five benchmark textual datasets, namely DBLP1, DBLP2 proposed in [5], GitHub-IA-Bio [16], Classic3 by Cornell University, and an extract of 8,000 texts of the AG-news[1] dataset. We have selected $v = 5$ text representations, namely BOW, Entity Embedding, Skipgram, XLNET, and Sentence-Bert (S-Bert). The feature of each dataset is presented in Table 1.

Table 1: Description of textual datasets.

		Documents	Clusters	Features				
				Bow	Entity	Skipgram	XLNET	S-BERT
Datasets	**DBLP1**	1919	3	6931	1980	100	120	384
	DBLP2	2223	3	2500	1835			
	GitHub	1528	2	4994	1643			
	Classic3	3891	3	23590	6920			
	Agnews	8000	4	22604	8314			

Two data representations of text were generated using the process described in the introduction namely feature representation $\mathbf{E}_b$ and similarity representation $\mathbf{X}_b$ for each of text representation $b = 1...v$. Finally, a tensor representation $\mathcal{X} \in \mathcal{R}^{n \times n \times v}$ that contains all similarity matrices of the v representations is constructed.

3.2 What is the impact of combining implicit and explicit consensus on text clustering performances?

We propose an evaluation with consensus clustering algorithms on feature matrices namely Consensus – GMM, Consensus – Kmeans, and Consensus – SKmeans (or ESK). We also use graph consensus clustering algorithms on similarity matrices namely, Consensus – CoclustMod, Consensus – CoclustInfo, and Consensus – SPLBM. We used the ClusterEnsembles[2] consensus algorithm to obtain the consensus partition. We used the TGM approach as an implicit consensus method.

We evaluate all algorithms in terms of Accuracy, normalized mutual information (NMI)[14], and Purity on the five datasets, using 30 random initializations, and the average value of each metric is reported in table 2.

Table 2: Comparison of consensus clustering results in terms of ACC, NMI, and Purity. The bold blue values represent the best performances. The bold ones are the second-best performances.

Data	Data Representation	Consensus	Algorithms	ACC	NMI	Purity
DBLP1	*Embedding*	EC	Consensus – GMM	0.564	0.149	0.568
			Consensus – Kmeans	0.706	0.302	0.706
			Consensus – SKmeans (ESK)	0.612	0.226	0.62
	Similarity	EC	Consensus – CoclustMod	0.595	0.224	0.609
			Consensus – CoclustInfo	0.626	0.238	0.632
			Consensus – SPLBM	0.616	0.24	0.622
		IC	TGM	0.571	0.218	0.571
	Emb + Sim	EC+IC	IEcons	**0.77**	**0.389**	**0.77**
DBLP2	*Embedding*	EC	Consensus – GMM	0.512	0.114	0.554
			Consensus – Kmeans	**0.63**	**0.19**	**0.63**
			Consensus – SKmeans (ESK)	0.554	0.177	0.579
	Similarity	EC	Consensus – CoclustMod	0.558	0.175	0.581
			Consensus – CoclustInfo	0.546	0.168	0.573
			Consensus – SPLBM	0.552	0.157	0.571
		IC	TGM	**0.591**	0.185	**0.591**
	Emb + Sim	EC+IC	IEcons	**0.63**	**0.254**	**0.63**
GitHub	*Embedding*	EC	Consensus – GMM	0.704	0.141	0.706
			Consensus – Kmeans	0.709	0.141	0.709
			Consensus – SKmeans (ESK)	0.77	0.246	0.77
	Similarity	EC	Consensus – CoclustMod	0.763	0.233	0.763
			Consensus – CoclustInfo	0.77	0.245	0.77
			Consensus – SPLBM	0.579	0.065	0.657
		IC	TGM	**0.859**	**0.402**	**0.859**
	Emb + Sim	EC+IC	IEcons	**0.986**	**0.93**	**0.986**
Classic3	*Embedding*	EC	Consensus – GMM	0.916	0.743	0.916
			Consensus – Kmeans	0.923	0.758	0.923
			Consensus – SKmeans (ESK)	0.919	0.749	0.919
	Similarity	EC	Consensus – CoclustMod	0.918	0.749	0.918
			Consensus – CoclustInfo	0.927	0.774	0.927
			Consensus – SPLBM	0.837	0.629	0.843
		IC	TGM	**0.984**	**0.919**	**0.984**
	Emb + Sim	EC+IC	IEcons	**0.986**	**0.93**	**0.986**
AG-news	*Embedding*	EC	Consensus – GMM	0.499	0.137	0.5
			Consensus – Kmeans	0.543	0.161	0.543
			Consensus – SKmeans (ESK)	0.55	0.194	0.551
	Similarity	EC	Consensus – CoclustMod	0.511	0.186	0.517
			Consensus – CoclustInfo	0.506	0.194	0.511
			Consensus – SPLBM	0.424	0.136	0.436
		IC	TGM	**0.612**	**0.382**	**0.617**
	Emb + Sim	EC+IC	IEcons	**0.656**	**0.413**	**0.656**

We observe that IEcons achieves overall the best results in terms of clustering performances based on Accuracy, NMI[14], and Purity metrics. This can be explained by the fact that IEcons optimizes the explicit and implicit consensus clustering simultaneously considering the two data representations namely feature and similarity

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

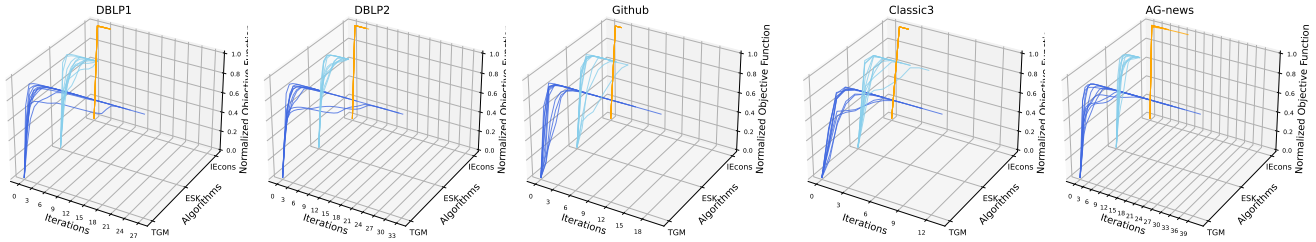

Figure 4: Evolution of the normalized objective function of IEcon, explicit and implicit consensus.

representations, allowing capturing the local and global similarities between texts. We also notice, that TGM achieves the second-best result which seems to match with the results obtained by the tensor clustering approach TSPLBM for implicit consensus in [6].

3.3 Is IEcons algorithm convergence better than explicit and implicit consensus clustering?

In this section we want to study the evolution of the normalized objective function of IEcons and compare it to implicit and explicit consensus clustering. For this end, Consensus − SKmeans algorithm (or ESK) is used as explicit consensus, and TGM algorithm for implicit consensus. Figure 4 represents the evolution of the normalized objective function of ESK, TGM, and IEcons on the five datasets using 30 random initializations.

Compared to ESK and TGM objective functions, IEcons runs are very similar over all iterations, which highlights the robustness and stability of the proposed algorithm. Also, IEcons's objective function converges quickly and reaches the highest normalized objective function value. We also observe that TGM is more stable compared to ESK but slower to achieve convergence due to the tensorial structure of the data.

3.4 How do dataset sizes impact IEcons clustering performances?

To answer this research question, we created different subsets of the AG-news dataset which is the largest among the five datasets presented previously. In our random sampling of AG-news, we made sure that the four clusters were present in all subsets. The size of the created datasets is in the range of 200 to 8000 with a step equals to 200, which represents 40 created sub-datasets.

We have run TGM and IEcon algorithms over the 40 sub-datasets created with 5 random initializations. Figure 5. presents the obtained results in terms of % of NMI improvement. Thus, the positive values give an advantage to IEcon and show that in major cases it outperforms the TGM algorithm for text clustering task. Also, we notice that there are very few cases where the values are negative, and when this happens these negative percentages are very low compared to the positive ones when the percentages are more important. Finally, even if the dataset size increases, the IEcon seems to be robust and stable regarding the data scalability.

4 CONCLUSION

This paper presented a new approach for simultaneous explicit and implicit consensus clustering named IEcons. The obtained results on five benchmark datasets show the effectiveness of IEcons in dealing with multi-text representations and different data representations.

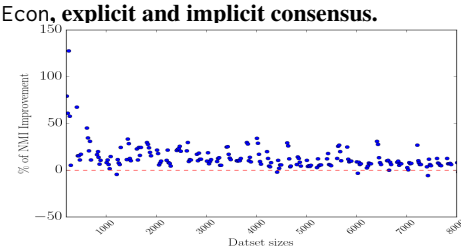

Figure 5: Percentage of NMI improvement for IEcons comparing to TGM on AG-news varying the dataset size.

We also proved that IEcons convergence is better than explicit and implicit consensus, and achieves promising results regarding data scalability. For future work, we plan to tackle the problem of cluster number selection of IEcons and extend IEcons to other applications such as image clustering.

REFERENCES

[1] Charu C. Aggarwal and ChengXiang Zhai. 2012. *A Survey of Text Clustering Algorithms*. Springer, 77–128.

[2] Melissa Ailem, François Role, and Mohamed Nadif. 2015. Co-clustering document-term matrices by direct maximization of graph modularity. In *CIKM*. 1807–1810.

[3] Melissa Ailem, François Role, and Mohamed Nadif. 2017. Sparse poisson latent block model for document clustering. *IEEE Transactions on Knowledge and Data Engineering* 29, 7 (2017), 1563–1576.

[4] Rafika Boutalbi, Mira Ait-Saada, Anastasiia Iurshina, Steffen Staab, and Mohamed Nadif. 2022. Tensor-based graph modularity for text data clustering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2227–2231.

[5] Rafika Boutalbi, Lazhar Labiod, and Mohamed Nadif. 2019. Sparse tensor co-clustering as a tool for document categorization. In *ACM SIGIR*. 1157–1160.

[6] Rafika Boutalbi, Lazhar Labiod, and Mohamed Nadif. 2021. Implicit consensus clustering from multiple graphs. *Data Mining and Knowledge Discovery* 35, 6 (2021), 2313–2340.

[7] Christian Buchta, Martin Kober, Ingo Feinerer, and Kurt Hornik. 2012. Spherical k-means clustering. *Journal of statistical software* 50, 10 (2012), 1–22.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* 39 (1977), 1–38.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ACL*. 4171–4186.

[10] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP*.

[14] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3 (2002), 583–617.

[15] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: Generalized Autoregressive Pretraining for*

Language Understanding.

[16] Yu Zhang, Frank F. Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. 2019. HiGitClass: Keyword-Driven Hierarchical Classification of GitHub Repositories. In *ICDM'19*. 876–885.