Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!

Suzanna Sia **Ayush Dalmia** **Sabrina J. Mielke**
Department of Computer Science
Johns Hopkins University
Baltimore, MD, USA
ssia1@jhu.edu, adalmia1@jhu.edu, sjmielke@jhu.edu

arXiv:2004.14914v2 [cs.CL] 6 Oct 2020

Abstract

Topic models are a useful analysis tool to uncover the underlying themes within document collections. The dominant approach is to use probabilistic topic models that posit a generative story, but in this paper we propose an alternative way to obtain topics: clustering pre-trained word embeddings while incorporating document information for weighted clustering and reranking top words. We provide benchmarks for the combination of different word embeddings and clustering algorithms, and analyse their performance under dimensionality reduction with PCA. The best performing combination for our approach performs as well as classical topic models, but with lower runtime and computational complexity.

1 Introduction

Topic models are the standard approach for exploratory document analysis (Boyd-Graber et al., 2017), which aims to uncover main themes and underlying narratives within a corpus. But in times of distributed and even contextualized embeddings, are they the only option?

This work explores an alternative to topic modeling by casting 'key themes' or 'topics' as *clusters of word types* under the modern distributed representation learning paradigm: unsupervised pre-trained word embeddings provide a representation for each word type as a vector, allowing us to cluster them based on their distance in high-dimensional space. The goal of this work is not to strictly outperform, but rather to benchmark standard clustering of modern embedding methods against the classical approach of Latent Dirichlet Allocation (LDA; Blei et al., 2003).

We restrict our study to influential embedding methods and focus on centroid-based clustering algorithms as they provide a natural way to obtain the top words in each cluster based on distance from the cluster center.[1]

Aside from reporting the best performing combination of word embeddings and clustering algorithm, we are also interested in whether there are consistent patterns: embeddings which perform consistently well across clustering algorithms might be good representations for unsupervised document analysis, clustering algorithms that perform consistently well are more likely to generalize to future word embedding methods.

To make our approach reliably work as well as LDA, we incorporate corpus frequency statistics directly into the clustering algorithm, and quantify the effects of two key methods, 1) weighting terms during clustering and 2) reranking terms for obtaining the top J representative words. Our contributions are as follows:

- We systematically apply centroid-based clustering algorithms on top of a variety of pre-trained word embeddings and embedding methods for document analysis.
- Through weighted clustering and reranking of top words we obtain sensible topics; the best performing combination is comparable with LDA, but with smaller time complexity and empirical runtime.
- We show that further speedups are possible by reducing the embedding dimensions by up to 80% using PCA.

2 Related Work and Background

Analyzing documents by clustering word embeddings is a natural idea—clustering has been used

[1] We found that using non-centroid-based hierarchical, or density based clustering algorithms like DBScan resulted in worse performance and more hyperparameters to tune.

for readability assessment (Cha et al., 2017), argument mining (Reimers et al., 2019), document classification and document clustering (Sano et al., 2017), *inter alia*. So far, however, clustering word embeddings has not seen much success for the purposes of topic modeling. While many modern efforts have attempted to *incorporate* word embeddings *into* the probabilistic LDA framework (Liu et al., 2015; Nguyen et al., 2015; Das et al., 2015; Zhao et al., 2017; Batmanghelich et al., 2016; Xun et al., 2017; Dieng et al., 2019), relatively little work has examined the feasibility of *clustering embeddings directly*.

Xie and Xing (2013) and Viegas et al. (2019) first cluster documents and subsequently find words within each cluster for document analysis. Sridhar (2015) targets short texts where LDA performs poorly in particular, fitting GMMs to learned word2vec representations. De Miranda et al. (2019) cluster using self-organising maps, but provide only qualitative results.

In contrast, our proposed approach is straightforward to implement, feasible for regular length documents, requires no retraining of embeddings, and yields qualitatively and quantitatively convincing results. We focus on centroid based k-means (KM), Spherical k-means (SK), and k-medoids (KD) for hard clustering, and von Mises-Fisher Models (VMFM) and Gaussian Mixture Models (GMM) for soft clustering; as pre-trained embeddings we consider word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017), Spherical (Meng et al., 2019), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018).

3 Methodology

After preprocessing and extracting the vocabulary from our training documents, each word type is converted to its embedding representation (averaging all of its tokens for contextualized embeddings; details in §5.3). Following this we apply the various clustering algorithms on the entire training corpus vocabulary to obtain k clusters, using weighted (§3.2) or unweighted word types. After the clustering algorithm has converged, we obtain the top J words (§3.1) from each cluster for evaluation. Note that one potential shortcoming of our approach is the possibility of outliers forming their own cluster, which we leave to future work.

Figure 1: The figure on the left shows the cluster center ($\star$) without weighting, while the figure on the right shows that after weighting (larger points have higher weight) a hopefully more representative cluster center is found. Note that top words based on distance from the cluster center could still very well be low frequency word types, motivating reranking (§3.3).

3.1 Obtaining top-J words

In traditional topic modeling (LDA), the top J words are those with highest probability under each topic-word distribution. For centroid based clustering algorithms, the top words of some cluster i are naturally those closest to the cluster center $c^{(i)}$, or with highest probability under the cluster parameters. Formally, this means choosing the set of types J as

$$\underset{J:\,|J|=10}{\operatorname{argmin}} \sum_{j \in J} \begin{cases} \|c^{(i)} - x_j\|_2^2 & \text{for KM/KD,} \\ \cos(c^{(i)}, x_j) & \text{for SK,} \\ f(x_j \mid c^{(i)}, \Sigma_i) & \text{for GMM/VMFM.} \end{cases}$$

Our results in §6 focus on KM and GMM, as we observe that k-medoids, spherical KM and von Mises-Fisher tend to perform worse than KM and GMM (see App. A, App. B).

Note that it is possible to extend this approach to obtain the top *topics given a document*: compute similarity scores between learned topic cluster centers and all word embeddings from that particular document, and normalize them using softmax to obtain a (non-calibrated) probability distribution.

Crucial to our method is the incorporation of corpus statistics on top of vanilla clustering algorithms, which we will describe in the remainder of this section.

3.2 Weighting while clustering

The intuition of *weighted clustering* is based on the formulation of classical LDA which models the probability of the word type t belonging to a topic i as $\frac{N_{t,i}+\beta_t}{\sum_{t'} N_{t'i}+\beta_{t'}}$, where $N_{t,i}$ refers to the number of times word type t has been assigned to topic i, and

β is a parameter of the Dirichlet prior on the per-topic word distribution. In our case, illustrated by the schematic in Fig. 1, weighting is a natural way to account for the frequency effects of vocabulary terms during clustering.

3.3 Reranking when obtaining topics

When obtaining the top-J words that make up a cluster's topic, we also consider *reranking* terms, as there is no guarantee that words closest to cluster centers are important word types. We will show in Table 2 that without reranking, clustering yields "sensible" topics but low NPMI scores.

3.4 Which corpus statistics?

To incorporate corpus statistics into the clustering algorithm, we examine three different schemes[2] to assign weights to word types, where n_t is the count of word type t in corpus D, and d is a document:

$$\textbf{tf} = \frac{n_t}{\sum_{t'} n_{t'}} \tag{1}$$

$$\textbf{tf-df} = \text{tf} \cdot \frac{|\{d \in D \mid t \in d\}|}{|D|} \tag{2}$$

$$\textbf{tf-idf} = \text{tf} \cdot \log\left(\frac{|D|}{|\{d \in D \mid t \in d\}| + 1}\right) \tag{3}$$

These scores can now be used for *weighting* word types when clustering ($\diamond^w$), *reranking* top 100 words ($\diamond_r$) after, both ($\diamond_r^w$), or neither (simply $\diamond$). We find that simply using **tf** outperforms the other weighting schemes (App. C). Our results and subsequent analysis in §6 uses **tf** for weighting and reranking.

4 Computational Complexity

The complexity of KM is $O(tknm)$, and of GMM is $O(tknm^3)$, for t iterations,[3] k clusters (topics), n word types (unique vocabulary), and m embedding dimensions. Weighted variants have a one-off cost of weight initialization, and contribute a constant factor when recalulculating the centroid during clustering. Reranking has an additional $O(n \cdot \log(n_k))$ factor, where n_k is the average number of elements in a cluster. In contrast, LDA via collapsed Gibbs sampling has a complexity of $O(tkN)$, where N is the number of all tokens, so when $N \gg n$, clustering methods can potentially achieve better performance-complexity tradeoffs.

Note that running ELMo and BERT over documents also requires iterating over all tokens, but only once, and not for every topic and iteration.

4.1 Cost of obtaining Embeddings

For readily available pretrained word embeddings such as word2vec, FastText, GloVe and Spherical, the embeddings can be considered as 'given' as the practioner does not need to generate these embeddings from scratch. However for contextual embeddings such as ELMo and BERT, there is additional computational cost in obtaining these embeddings before clustering, which requires passing through RNN and transformer layers respectively. This can be trivially parallelised by batching the context window (usually a sentence). We use standard pretrained ELMo and BERT models in our experiments and therefore do not consider the runtime of training these models from scratch.

5 Experimental Setup

Our implementation is freely available online.[4]

5.1 Datasets

We use the 20 newsgroup dataset (20NG) which contains around 18000 documents and 20 categories,[5] and a subset of Reuters21578[6] which contains around 10000 documents.

5.2 Evaluation (Topic Coherence)

We adopt a standard 60-40 train-test split for 20NG and 70-30 for Reuters.

The top 10 words (§3.1) were evaluated using *normalized pointwise mutual information* (NPMI; Bouma, 2009) which has been shown to correlate with human judgements (Lau et al., 2014). NPMI ranges from $[-1, 1]$ with 1 indicating perfect association. The train split is used to obtain the top topic words in an unsupervised fashion (we do not use any document labels), and the test split is used to evaluate the "topic coherence" of these top words. NPMI scores are averaged across all topics.

For both datasets we use 20 topics; which gives best NPMI out of 20, 50, 100 topics for Reuters, and is the ground truth number for 20NG. The

[2] We also experimented with various scaling methods such as robust scaling, logistic-sigmoid, and log transform but found that these do not improve performance.

[3] In general, t required for convergence differs for clustering algorithm and embedding representation. However we can specify the maximum number of iterations as a constant factor for worst case analysis.

[4] https://github.com/adalmia96/Cluster-Analysis

[5] http://qwone.com/~jason/20Newsgroups/

[6] https://www.nltk.org/book/ch02.html

	Reuters								*20 Newsgroups*							
	$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$		$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$	
	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM
Word2vec	-0.39	-0.47	-0.21	-0.09	0.02	0.01	0.03	0.08	-0.21	-0.10	-0.11	0.13	0.18	0.16	0.19	0.20
ELMo	-0.73	-0.55	-0.43	0.00	-0.10	-0.08	-0.02	0.06	-0.56	-0.13	-0.38	0.18	0.13	0.14	0.16	0.19
GloVe	-0.67	-0.59	-0.04	0.01	-0.27	-0.03	0.01	0.05	-0.18	-0.12	0.06	0.24	0.22	0.23	0.23	0.23
Fasttext	-0.68	-0.70	-0.46	-0.08	0.00	0.00	0.06	0.11	-0.32	-0.20	-0.18	0.21	0.24	0.23	0.25	0.24
Spherical	-0.53	-0.65	-0.07	0.09	0.01	-0.05	0.10	0.12	-0.05	-0.24	0.24	0.23	0.25	0.22	0.26	0.24
BERT	-0.43	-0.19	-0.07	0.12	0.00	-0.01	0.12	0.15	0.04	0.14	0.25	0.25	0.17	0.19	0.25	0.25
average	-0.57	-0.52	-0.21	0.01	-0.06	-0.03	0.05	0.10	-0.21	-0.11	-0.02	0.21	0.20	0.20	0.23	0.23
std. dev.	0.14	0.18	0.19	0.09	0.12	0.03	0.05	0.04	0.21	0.13	0.25	0.05	0.04	0.04	0.04	0.02

Table 1: NPMI Results (higher is better) for pre-trained word embeddings and k-means (KM), and Gaussian Mixture Models (GMM). $\diamond^w$ indicates weighted and $\diamond_r$ indicates reranking of top words. For Reuters (left table), LDA has an NPMI score of 0.12, while GMM_r^w BERT achieves 0.15. For 20NG (right), both LDA and KM_r^w Spherical achieve a score of 0.26. All results are averaged across 5 random seeds.

NPMI scores presented in Table 1 are averaged across cluster centers initialized using 5 random seeds.

5.3 Preprocessing

We lowercase tokens, remove stopwords, punctuation and digits, and exclude words that appear in less than 5 documents and appear in long sentences of more than 50 words, removing email artifacts and noisy token sequences which are not valid sentences. An analysis on the effect of rare word removal can be found in §6.2.

For contextualized word embeddings (BERT and ELMo), sentences served as the context window to obtain the token representations. Subword representations were averaged for BERT, which performs better than just using the first subword.

6 Results and Discussion

Our main results are shown in Table 1.

6.1 Runtime

Running LDA with `MALLET` (McCallum, 2002) takes a minute, but performs no better than KM_r^w, which takes little more than 10 seconds on CPU using `sklearn` (Pedregosa et al., 2011), and 3-4 seconds using a simple implementation using JAX (Bradbury et al., 2018) on GPU.

6.2 Weighting

From Table 1, we see that reranking and weighting greatly improves clustering performance across different embeddings. As a first step to uncover why, we investigate how sensitive our methods are to restricting the clustering to only *frequently appearing word types*. Visualized in Fig. 3, we find that as we vary the cutoff term frequency, thus changing the vocabulary size and allowing more rare words on the x-axis, NPMI is more affected for the models without reweighting. This suggests that reweighting using term frequency is effective for clustering without the need for ad-hoc restriction of infrequent terms—without it, all combinations perform poorly compared to LDA. In general, GMM outperforms KM for both weighted and unweighted variants averaged across all embedding methods ($p < 0.05$).[7]

BERT (topic12)		Spherical (topic19)	
KM	KM_r	KM	KM_r
vram	drive	detector	earth
vesa	hard	electromagnetic	nasa
cmos	card	magnetic	satellite
portable	computer	spectrometer	orbit
micron	chip	infrared	surface
nubus	machine	optical	energy
digital	video	velocity	radar
machine	hardware	radiation	solar
motherboards	clipper	solar	spacecraft
hardware	controller	telescope	electrical
NPMI: -0.36	*NPMI:* **0.15**	*NPMI:* -0.01	*NPMI:* **0.36**

Table 2: Top 10 words in a topic on 20NG and overall NPMI, for k-means (KM) before and after reranking (KM_r): reranking clearly improves NPMI for BERT and Spherical.

6.3 Reranking

For KM, extracted topics before reranking results in reasonable looking themes, but scores poorly on NPMI. Reranking strongly improves KM on average ($p < 0.02$) for both Reuters and 20NG. Examples before and after reranking are provided in Table 2. This indicates that while cluster centers are centered around valid themes, they are surrounded by low frequency word types.

[7] Two-tailed t-test for GMM^w vs KM^w.

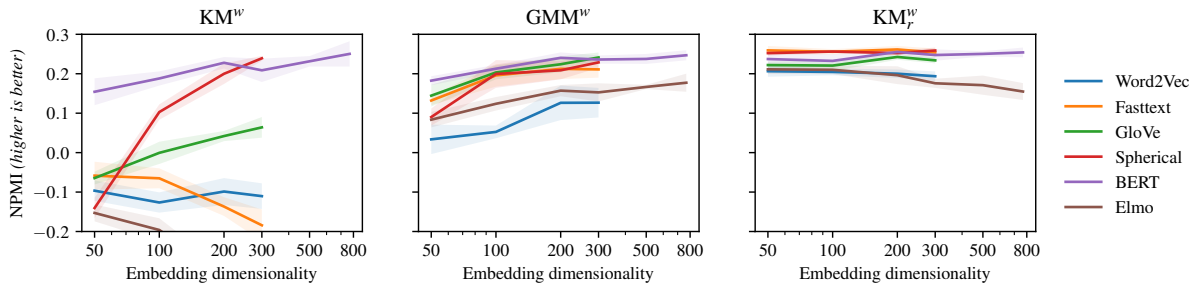

Figure 2: Plots showing the effect of PCA dimension reduction on different embedding and clustering algorithms. KM^w_r which we advocate over GMMs for efficiency, allows for dimension reduction of up to 80%.

We observe that when applying reranking to GMM^w the gains are much less pronounced than KM^w. The top topic words before and after reranking for BERT-GMM^w have an average Jaccard similarity score of 0.910, indicating that the cluster centers learned by weighted GMMs are already centered at word types of high frequency in the training corpus.

6.4 Embeddings

Spherical embeddings and BERT perform consistently well across both datasets. For 20NG, KM^w_r Spherical and LDA both achieve 0.26 NPMI. For Reuters, GMM^w_r BERT achieves the top NPMI score of 0.15 compared to 0.12 of LDA. Word2vec and ELMo (using only the last layer[8]) perform poorly compared to the other embeddings. FastText and GloVe can achieve similar performance to BERT on 20NG but are slightly inferior on Reuters.

Training or fine-tuning embeddings on the given data prior to clustering could potentially achieve better performance, but we leave this to future work.

6.5 Qualitative results

We find that our approach yields a greater diversity within topics as compared to LDA while achieving comparable coherence scores (App. D). Such topics are arguably more valuable for exploratory analysis.

6.6 Dimensionality Reduction

We apply PCA to the word embeddings before clustering to investigate the amount of redundancy in the dimensions of large embeddings, which impact clustering complexity (§4). With reranking, the dimensions of all embeddings can be reduced by more than 80% (Fig. 2).

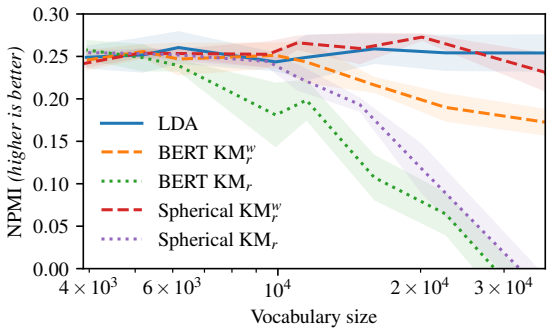

Figure 3: NPMI as a function of vocabulary size reduced by term frequency on 20NG. Embeddings are more sensitive to noisy vocabulary (infrequent terms) than LDA, but reweighting ($\diamond^w$) helps to alleviate this.

We observe that KM^w_r can consistently reduce the number of dimensions across different embedding types without loss of performance. Although GMM^w does not require reranking for good performance, it's cubic complexity indicates that KM^w_r might be preferred in practical settings.

7 Conclusion

We outlined a methodology for clustering word embeddings for unsupervised document analysis, and presented a systematic comparison of various influential embedding methods and clustering algorithms. Our experiments suggest that pretrained word embeddings (both contextualized and non-contextualized), combined with tf-weighted k-means and tf-based reranking, provide a viable alternative to traditional topic modeling at lower complexity and runtime.

[8]Selected as best performing by manually testing 13 different mixing ratios.

Acknowledgments

We thank Aaron Mueller, Pamela Shapiro, Li Ke, Adam Poliak, Kevin Duh and the anonymous reviewers for their feedback.

References

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2016, page 537. NIH Public Access.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. 2018. JAX: composable transformations of Python+NumPy programs.

Miriam Cha, Youngjune Gwon, and HT Kung. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2003–2006.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. Http://mallet.cs.umass.edu.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *Advances in Neural Information Processing Systems*, pages 8206–8215.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Guilherme Raiol de Miranda, Rodrigo Pasti, and Leandro Nunes de Castro. 2019. Detecting topics in documents by clustering word vectors. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 235–243. Springer.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. *arXiv preprint arXiv:1906.09821*.

Motoki Sano, Austin J Brockmeier, Georgios Kontonatsios, Tingting Mu, John Y Goulermas, Jun'ichi Tsujii, and Sophia Ananiadou. 2017. Distributed document and phrase co-embeddings for descriptive clustering. In *Proceedings of the 15th Conference of*

the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 991–1001.

Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised topic modeling for short texts using distributed representations of words. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 192–200.

Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. Cluwords: exploiting semantic word clustering representation for enhanced topic modeling. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 753–761.

Pengtao Xie and Eric P Xing. 2013. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874*.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *IJCAI*, pages 4207–4213.

He Zhao, Lan Du, and Wray Buntine. 2017. A word embeddings informed focused topic model. In *Asian Conference on Machine Learning*, pages 423–438.

A k-means (KM) vs k-medoids (KD)

To further understand the effect of other centroid based algorithms on topic coherence, we also applied the k-medoids (KD) clustering algorithm. KD is a hard clustering algorithm similar to KM but less sensitive to outliers.

As we can see in Table 3, in all cases KD usually did as well or worse than KM. KD also did relatively poorly after frequency reranking. Where KD did do better than KM, the difference is not very striking and the NPMI scores were still quite below the other top performing models.

B Results for Spherical k-means and Von Mises-Fisher Mixture

Table 4 shows the overall bad performance of spherical clustering methods, specifically Spherical k-Means (SKM) and von-Mises-Fisher mixtures (VMFM).

C Comparing Different Reranking Schemes

We present the results for using different reranking schemes for KM (Table 5) and Weighted KM for Frequency (Table 6).

We can see that compared to the TF results in the main paper, other schemes for reranking such as aggregated TF-IDF and TF-DF improve over the original hard clustering, but fare worse in comparison with reranking with TF.

D Qualitative Comparison of Topics Generated

We present the different topics generated using LDA (Table 7) and topics generated using BERT KM_r^w for the Reuters dataset (Table 8). Note that

	KM	KD	KM_r	KD_r
Word2Vec	-0.21	-0.32	0.18	0.12
FastText	-0.33	-0.39	0.24	0.19
GloVe	-0.18	-0.43	0.22	0.08
BERT	0.04	-0.06	0.17	0.15
ELMo	-0.56	-0.56	0.13	0.12
Spherical	-0.05	-0.07	0.25	0.22
average	-0.22	-0.31	0.20	0.15
std. dev.	0.21	0.20	0.04	0.05

Table 3: Results for pre-trained word embeddings and k-means (KM) and k-medoids (KD). r indicates reranking of top words using term frequency.

	Reuters							
	$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$	
	SKM	VMFM	SKM	VMFM	SKM	VMFM	SKM	VMFM
Word2vec	-0.70	-0.85	-0.43	-0.88	-0.16	-0.05	-0.19	-0.05
ELMo	-0.74	-0.88	-0.37	-0.87	-0.14	-0.10	0.00	-0.12
GloVe	-0.52	-0.88	-0.11	-0.88	0.00	-0.18	0.06	-0.17
Fasttext	-0.85	-0.89	-0.65	-0.87	-0.18	-0.08	-0.18	-0.10
Spherical	-0.50	-0.81	-0.08	-0.82	0.01	-0.07	0.10	-0.09
BERT	-0.40	-0.88	-0.06	-0.65	-0.03	-0.14	0.11	-0.16
average	-0.62	-0.87	-0.28	-0.83	-0.08	-0.10	-0.02	-0.12
std. dev.	0.17	0.03	0.24	0.09	0.09	0.05	0.14	0.04

	20 Newsgroups							
	$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$	
	SKM	VMFM	SKM	VMFM	SKM	VMFM	SKM	VMFM
Word2vec	-0.37	-0.59	-0.17	-0.88	0.15	0.17	0.14	0.16
ELMo	-0.52	-0.66	-0.30	-0.87	0.16	0.10	0.20	0.12
GloVe	0.00	-0.62	0.23	-0.88	0.25	0.13	0.24	0.14
Fasttext	-0.60	-0.58	-0.26	-0.54	0.12	0.19	0.14	0.19
Spherical	-0.04	-0.54	0.22	-0.82	0.25	0.22	0.25	0.21
BERT	0.06	-0.62	0.22	-0.65	0.23	0.11	0.25	0.10
average	-0.24	-0.60	-0.01	-0.77	0.19	0.15	0.20	0.15
std. dev.	0.29	0.04	0.26	0.14	0.06	0.05	0.05	0.04

Table 4: NPMI Results (higher is better) for pre-trained word embeddings and Spherical k-means (SKM), and von Mises-Fisher Mixtures (VMFM). $\diamond^w$ indicates weighted and $\diamond_r$ indicates reranking of top words.

	TF	TF-IDF	TF-DF
Word2Vec	0.18	0.15	0.17
FastText	0.24	0.23	0.23
GloVe	0.22	0.17	0.21
BERT	0.17	0.15	0.17
ELMo	0.13	0.09	0.14
Spherical	0.25	0.22	0.24
average	0.20	0.17	0.19
std. dev.	0.04	0.05	0.04

Table 5: Results for k-means (without weighting) with pre-trained word embeddings using different reranking metrics : TF, TF-IDF, and TF-DF.

	$\diamond^w$ TF	$\diamond^w$ TF-IDF	$\diamond^w$ TF-DF
Word2Vec	0.19	0.17	0.20
FastText	0.25	0.25	0.25
GloVe	0.23	0.21	0.23
BERT	0.25	0.24	0.25
ELMo	0.16	0.15	0.16
Spherical	0.26	0.24	0.25
average	0.23	0.21	0.22
std. dev.	0.04	0.04	0.04

Table 6: Results for k-means (weighted) pre-trained word embeddings using different reranking metrics: TF, TF-IDF and TF-DF weighted with term frequency.

unlike LDA, which uses the highest posterior probability allowing duplicate words to appear in duplicate topics, using a hard clustering algorithm for assignment mean that each word is assigned to one topic only. We can see compared to the LDA topics which tend to contain topics mostly regarding wealth and profits, clustering with BERT KM_r^w introduces new topics in involving locations and corporate positions. We see overall that using clustering allows for a discovery for a greater diversity of topics due to the greater diversity of words over all the topics.

Top 10 Word for Each Topic	NPMI
dollar rate rates exchange currency market dealers central interest point	0.369
year growth rise government economic economy expected domestic inflation report	0.355
gold reserves year tons company production exploration ounces feet mine	0.290
billion year rose dlrs fell marks earlier figures surplus rise	-0.005
year tonnes crop production week grain sugar estimated expected area	0.239
dlrs company sale agreement unit acquisition assets agreed subsidiary sell	-0.043
bank billion banks money interest market funds credit debt loans	0.239
tonnes wheat export sugar tonne exports sources shipment sales week	0.218
plan bill industry farm proposed government administration told proposal change	0.212
prices production price crude output barrels barrel increase demand industry	0.339
group company investment stake firm told companies capital chairman president	0.191
trade countries foreign officials told official world government imports agreement	0.298
offer company shares share dlrs merger board stock tender shareholders	0.074
shares stock share common dividend company split shareholders record outstanding	0.277
dlrs year quarter earnings company share sales reported expects results	-0.037
market analysts time added long analyst term noted high back	0.316
coffee meeting stock producers prices export buffer quotas market price	0.170
loss dlrs profit shrs includes year gain share mths excludes	-0.427
spokesman today government strike union state yesterday workers officials told	0.201
program corn dlrs prior futures price loan contract contracts cents	-0.287

Table 7: NPMI Scores and Top 10 words for the topics generated using LDA for the Reuters dataset

Top 10 Word for Each Topic	NPMI
rise increase growth fall change decline drop gains cuts rising	0.238
president chairman minister house baker administration secretary executive chief washington	0.111
make continue result include reduce open support work raise remain	0.101
january march february april december june september october july friday	0.043
year quarter week month earlier months years time period term	0.146
rose fell compared reported increased estimated revised adjusted unchanged raised	0.196
today major made announced recent full previously strong final additional	0.125
share stock shares dividend common cash stake shareholders outstanding preferred	0.281
dlrs billion tonnes marks francs barrels cents tonne barrel tons	-0.364
sales earnings business operations companies products markets assets industries operating	0.115
sale acquisition merger sell split sold owned purchase acquire held	0.003
board meeting report general commission annual bill committee association council	0.106
loss profit revs record note oper prior shrs gain includes	0.221
company corp group unit firm management subsidiary trust pacific holdings	0.058
prices price current total lower higher surplus system high average	0.198
offer agreement agreed talks tender plan terms program proposed issue	0.138
bank trade market rate exchange dollar foreign interest rates banks	0.327
told official added department analysts officials spokesman sources statement reuters	0.181
production export exports industry wheat sugar imports output crude domestic	0.262
japan government international world countries american japanese national states united	0.251

Table 8: NPMI Scores and Top 10 words for the topics generated using BERT KM_r^w for the Reuters dataset