Subtopic Clustering with a Query-Specific Siamese Similarity Metric

Anonymous ACL submission

Abstract

We propose a Query-Specific Siamese Similarity Metric (QS3M) for query-specific clustering of text documents. It uses fine-tuned BERT embeddings and trains a non-linear projection into a query-specific similarity space. We build on the idea of Siamese networks but include a third component, a representation of the query. The empirical evaluation for clustering employs two TREC datasets with two different clustering benchmarks each. When used to obtain query-relevant clusters, QS3M achieves a 12% performance improvement over a recently published BERT-based reference method and significantly outperforms other unsupervised baselines.

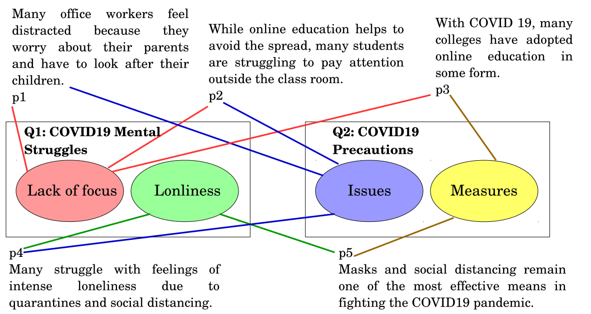


Figure 1: Different queries require different clusterings: for query Q1, "Covid19 Mental Struggles", the subtopics "Lack of Focus" and "Loneliness" are more relevant than clusters about "Issues" vs. "Measures"—and vice versa for query Q2. Cluster names are for illustration only.

1 Introduction

Users with conscious information needs (Taylor, 2015) tend to ask vague and under-specified queries, reflecting that the user does not know enough about a topic to phrase a concrete question. To answer such vague information needs, retrieval systems aim to cover as many relevant subtopics about the query as possible and provide the user with a comprehensive overview about the topic (Drosou and Pitoura, 2010). Explicit clustering is used as a separate post-processing step, to organize the retrieved results in topical groups such as for taxonomic browsing.[1] We however envision subtopic clustering to be a central component of a "retrieve-and-generate" system. Upon submission of a query, such system seamlessly retrieves relevant passages from the web and arranges them according to the subtopic clusters to generate a coherent, maximally relevant article for presentation to the user. In this work, we focus on the central step in this envisioned system: subtopic clustering.

Task Statement. Given a query q and a relevant set of passages[2] $\mathcal{P}_q$ which could be retrieved by a search system, our goal is to cluster passages in $\mathcal{P}_q$ into query-relevant subtopics.

A canonical approach to text clustering is to represent passages as vectors which govern a clustering algorithm through a similarity metric (e.g. TFIDF with cosine similarity) (Huang, 2008). Recently, neural embeddings and trained similarity functions obtain better performance (Xu et al., 2015; Reimers and Gurevych, 2019). However, an issue of such clustering approaches is that the similarity score between the two passages does not incorporate the query. We hypothesize that more relevant subtopic clusters can be found with a query-specific similarity metric. Even if the same set of passages are relevant for two queries, these would require different ways of clustering as illustrated in Figure 1. To address this, we design a query-specific text similarity metric, which when used with a clustering algorithm, will lead to query-specific clusters of retrieved passages.

Contribution. We develop a trainable query-specific similarity metric for text passages. The similarity metric is optimized to predict similarity scores that agree with the ground truth of passage clusters in the training data.

[1] https://www.yippy.com/

[2] The method can also be used with sentences, documents.

2 Related Work

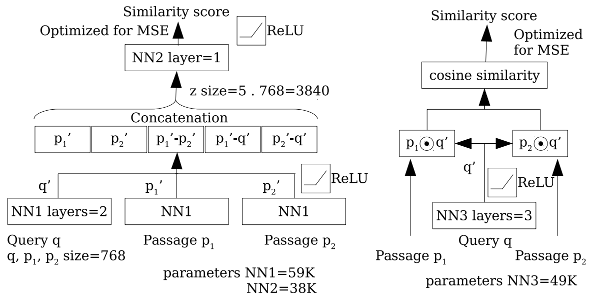
Previous work on text clustering (Gomaa and Fahmy, 2013; Bilenko et al., 2004; Metzler et al., 2007; Banea et al., 2012, *inter alia*) focuses on unsupervised lexical similarity metrics and their combinations. For semi-supervised clustering, Basu et al. (2002) have found pairwise binary constraints also known as *"must link"* and *"cannot link"* to be particularly effective. Query-specific clustering can be addressed as a separate step after retrieval, such as the extraction and co-occurrence analysis of keyphrases. Leung et al. (2008) uses information from the user's profile. Raiber and Kurland (2013) uses canopy clustering for re-rankings. Bernardini et al. (2009) uses keyphrases to identify clusters. Detailed study of search result clustering are available in the works of Carpineto et al. (2012) and Drosou and Pitoura (2010).

Clustering algorithms depend on a meaningful representation of text. Most lexical similarity metrics employ term-based vector representation of text such as TFIDF. Probabilistic topic models such as latent Dirichlet allocation (Blei et al., 2003) use the topic distribution to represent documents. With the advent of Transformer-based neural networks (Vaswani et al., 2017; Devlin et al., 2018), text embeddings have given rise to strong linguistic models. Zhang et al. (2019) study how to utilize the information captured at various layers of transformer networks for representing text. Reimers and Gurevych (2019) show how to fine-tune BERT for sentence clustering. This is an example of a trained similarity metric in which the query influences the candidate set but not the metric itself.

Research on query-specific clustering suggests that query information helps clustering. Recent Transformer-based embedding models have been demonstrated to capture high-quality topical information, but it is yet to be studied how to incorporate the query in such trainable embedding vector space that benefits query-specific clustering.

3 Approach

We focus on training a query-specific similarity metric between semantic representations of text passages, which is used in a distance-based clustering algorithm. Our rationale is that an ideal query-specific similarity metric should identify the query-relevant subtopics and ignore other spurious topical dimensions. For example in Figure 1, it should emit high similarity scores between passages discussing aspetcs of "Lack of Focus" in context of the query "Covid19 Mental Struggles". In this work, we assume that both query and passages are represented as vectors generated by a pre-trained embedding model.[3]



(a) QS3M Architecture. (b) QSS Architecture.

Figure 2: Model architectures. $a \odot b$ denotes element-wise multiplication (hadamard product).

Our similarity metric is designed to fit in the following clustering pipeline:

Step 1: A model is trained to predict the query-specific similarity between a pair of passages p_1, p_2 given a query q. **Step 2:** Given a query set Q and retrieved passage sets $\mathcal{P}_q$ for each query $q \in Q$, we apply the model to predict similarity scores between all passages in $\mathcal{P}_q$. **Step 3:** Given a set of query-specific similarities between passages in $\mathcal{P}_q$, we generate k_q clusters of passages for each query q with average-link agglomerative clustering.

The result of this pipeline are subtopic clusters that coincide with query-specific subtopics. Since it is an open question how to set the true number of cluster k_q, we omit this question in this work and assume that the number of clusters k_q is provided during evaluation.

Our central contribution in this work is the neural model used in Step 1 for query-specific similarity metric for passages, detailed in the following.

Query-Specific Siamese Similarity Metric (QS3M). Our goal is to, given a query q and a set of retrieved passages $\mathcal{P}_q$, model the similarity metric ϕ, where $\phi_q(p_i, p_j)$ denotes the similarity score between a pair of passages p_i, p_j from $\mathcal{P}_q$.

In the Query-Specific Siamese Similarity Metric (QS3M), we assume that the metric ϕ should model the complex interdependence between query and passages. This is captured by a siamese neural network with a third component for the query, inspired by the model proposed by Zeghidour et al.

[3]We use Sentence-BERT embedding vectors of size 768.

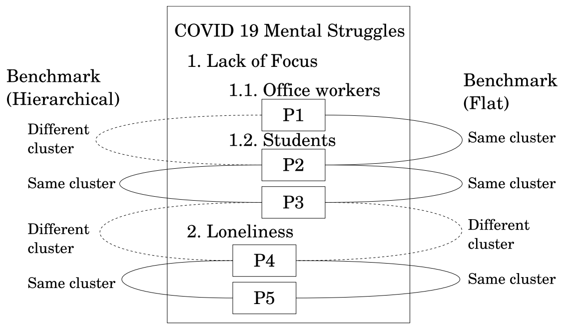


Figure 3: A train/test benchmark for query-specific clustering can be derived from source articles with sectioned outlines.

(2016). We implement ϕ using the neural architecture presented in Figure 2a. The fully-connected neural layer NN1 projects the query vector q and the pair of passage vectors p_1, p_2 into a different latent vector space that is more suitable for the query-specific similarity. To model the similarity, we observe how the pair of passages interact with the query as well as with each other in this transformed vector space. To formulate this three-way interaction, we concatenate three difference vectors, $p_1' - q', p_2' - q', p_1' - p_2'$ along with the projected passage vectors p_1', p_2' and obtain the vector z. Encapsulating these different interactions in a single vector allows subsequent neural layers to directly learn the complex relations between passage pairs in context of a query. A second neural layer NN2 operates as a binary classifier and from the vector z as input, predicts whether the pair of passages p_1, p_2 should share the same cluster or not. The neural layer NN2 is optimized for mean squared error loss.

Query-Specific Scaling metric (QSS). One may argue that we merely need to apply certain reweighting of passage representations to arrive at a query-specific similarity. The Query-Specific Scaling metric (QSS) is based on this assumption and models the similarity metric ϕ through learning a scaling vector q' that reweights passage vectors as depicted in Figure 2b.

Generating training data. To train and evaluate the similarity metric, we derive a benchmark where for given queries, pairs of passages are labeled with "same cluster" or "different cluster". Such a benchmark is derived from a corpus of articles where each article is relevant for a search query and each section of the article describes one subtopic as described in Figure 3. The *hierarchical* benchmark considers sub-hierarchy of each section of the article as separate topics whereas the *flat* benchmark only considers the top-level sections. Because the predominant number of pairs labeled as "different cluster" can negatively impact the training result, we balance the training dataset by sampling negative pairs. In order to reduce ambiguity for the *hierarchical* clustering benchmarks, we omit pairs from our training data when one passage in the pair is the parent and the other passage is in its child cluster. In this work, we derive a benchmark from Wikipedia articles, but our methods can be applied to other benchmarks as well.

4 Evaluation

We use the publicly available TREC Complex Answer Retrieval[4] (CAR) (Dietz et al., 2017) dataset version 2.0 of CAR year 1 for training and evaluation. We choose Sentence-BERT (Reimers and Gurevych, 2019) to generate the embedding vectors for query representations and passages. Sentence-BERT is pre-trained using training data obtained from 1.6 million queries in `train.v2.0`[5] with maximum input sequence length of 512. We also experiment with raw BERT embeddings without the pre-training step but observe that this degrades performance. The remaining queries from the `train.v2.0` are used to construct the training dataset for our models in *flat* benchmark style as described in Section 3. For evaluation, we use `benchmarkY1test` (referred to as *CAR-A*, 125 queries) and `benchmarkY1train` (*CAR-B*, 115 queries) from the CAR dataset. On average there are 7 true clusters per query for *flat* and 16 for *hierarchical* benchmarks.

Query representations. We explore the following options for representing the query.

- **Title (T):** Embedding of the article title.
- **Description (D):** Embedding of the introductory passage of the article (omitted from the passage set $\mathcal{P}_q$).
- **Passages (P):** The average of embeddings of all passages in the set $\mathcal{P}_q$.

Depending on which query representation we chose during training, we obtain three variations each of QS3M and QSS (e.g. QS3M with title query representations QS3M-T).

[4] http://trec-car.cs.unh.edu/

[5] We refer to filenames used in the CAR data set.

Table 1: Clustering performance in macro averaged Adjusted RAND index and paired t-test ($\alpha = 0.05$) with respect to SBERT-euc which is marked with $\star$. Significantly higher ▲ or lower ▼ methods according to paired t-test. Baseline methods are at the bottom.

	Flat		Hierarchical	
Methods	CAR-A	CAR-B	CAR-A	CAR-B
QS3M-P	0.300▲	0.307	0.237▲	0.276▲
QS3M-D	0.298▲	0.323▲	0.233	0.274▲
QS3M-T	0.289▲	0.306	0.217	0.246
QSS-P	0.249▼	0.295	0.219	0.226
QSS-D	0.263	0.304	0.221	0.255
QSS-T	0.269	0.296	0.225	0.239
QS3M-no-q	0.284▲	0.297	0.218	0.241
SBERT-euc	0.263$\star$	0.295$\star$	0.214$\star$	0.239$\star$
SBERT-cos	0.258	0.287	0.216	0.236
TFIDF-cos	0.071▼	0.068▼	0.109▼	0.120▼
Topic Model	≈ 0▼	0.009▼	≈ 0▼	≈ 0▼

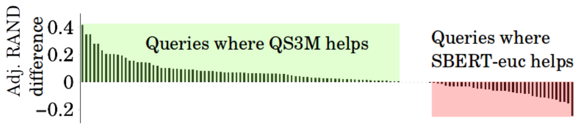


Figure 4: Helps-Hurts analysis: QS3M-P (top/green) vs. SBERT-euc (bottom/red) on *CAR-A flat*.

Baselines. We compare our methods to the following baselines:

- **SBERT-euc:** Euclidean distance of Sentence-BERT embedded passage vectors without any query-specific training.
- **SBERT-cos:** Same as SBERT-euc, but using the cosine similarity.
- **TFIDF-cos:** Cosine similarity between TFIDF vectors of passages.
- **Topic model:** Jensen-Shannon divergence between the topic distribution of two passages, estimated using LDA topic model with 200 topics (Blei et al., 2003). The topic model is trained on our training set.

Experimental results. We evaluate to which extent the query-specific similarity metric give rise to better clustering results on both *flat* and *hierarchical* clustering benchmarks of *CAR-A* and *CAR-B* datasets. We report the clustering results in Table 1 in terms of the macro-averaged Adjusted RAND index as a measure of clustering quality.[6]

It is evident from Table 1 that our approach of incorporating query information into the similarity metric leads to better clustering performance. For both *CAR-A* and *CAR-B*, QS3M achieves statistically significant improvements with respect to both clustering benchmarks. In particular, QS3M-P is the best performing method, achieving on average 12% relative improvement over the best performing baseline method, SBERT-euc. Also, QS3M without any query representation (QS3M-no-q) is worse than any other QS3M variant suggesting that to achieve a consistent improvement it is instrumental to train a query-specific similarity metric. In contrast, the simpler QSS model performs only on par with SBERT-euc.

We observe that query representations, *description* (D) and *passages* (P), achieve better results than *title* (T). We believe this is because the query titles only contain a few keywords which are not enough to capture useful context information.

We observe a large variance of clustering scores across queries. Hence, we perform a helps-hurts analysis on *CAR-A flat* presented in Figure 4 to compare the clustering performance of QS3M-P with the SBERT-euc baseline on a per-query basis. We find that for two-thirds of 125 *CAR-A* queries, QS3M-P obtains a better adjusted RAND index.

We note that the *hierarchical* benchmark has more true clusters than the *flat* benchmark. Furthermore, many *hierarchical* true clusters have only three or fewer passages. These attributes make the *hierarchical* dataset much more challenging to cluster. In spite of that we see similar improvements achieved by QS3M over SBERT baselines.

5 Conclusion

Our work is motivated by the hypothesis that subtopic clustering is influenced by the current query context and consequently a query-specific similarity metric is better suited for subtopic clustering. We propose Query-Specific Siamese Similarity Metric (QS3M) that provides empirical evidence in support of our hypothesis. Empirical evaluations demonstrate that subtopic clustering results can be improved by 12% with our proposed method over Sentence-BERT, a strong BERT-based method that does not take the query into account. We also find that long and descriptive query representations are more suitable in terms of clustering performance. While we envision QS3M to extract subtopics for automatic article generation, it can be applied to any context-specific text clustering task, such as domain-specific taxonomy extraction or search result diversification.

[6] Dataset and code will be released upon acceptance.

References

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 635–642. Association for Computational Linguistics.

Sugato Basu, Arindam Banerjee, and Raymond Mooney. 2002. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. Citeseer.

Andrea Bernardini, Claudio Carpineto, and Massimiliano D'Amico. 2009. Full-subtopic retrieval with keyphrase-based search results clustering. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 206–213. IEEE.

Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM.

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

Claudio Carpineto, Massimiliano D'Amico, and Giovanni Romano. 2012. Evaluating subtopic retrieval methods: Clustering versus diversification of search results. *Information Processing & Management*, 48(2):358–373.

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*.

Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. Trec complex answer retrieval overview. In *TREC*.

Marina Drosou and Evaggelia Pitoura. 2010. Search result diversification. *ACM SIGMOD Record*, 39(1):41–47.

Wael H Gomaa and Aly A Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18.

Anna Huang. 2008. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56.

Kenneth Wai-Ting Leung, Wilfred Ng, and Dik Lun Lee. 2008. Personalized concept-based clustering of search engine queries. *IEEE transactions on knowledge and data engineering*, 20(11):1505–1518.

Donald Metzler, Susan Dumais, and Christopher Meek. 2007. Similarity measures for short segments of text. In *European conference on information retrieval*, pages 16–27. Springer.

Fiana Raiber and Oren Kurland. 2013. Ranking document clusters using markov random fields. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 333–342.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Robert S Taylor. 2015. Question-negotiation and information seeking in libraries. *College & Research Libraries*, 76(3):251–267.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Neil Zeghidour, Gabriel Synnaeve, Nicolas Usunier, and Emmanuel Dupoux. 2016. Joint learning of speaker and phonetic similarities with siamese networks. In *INTERSPEECH*, pages 1295–1299.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.