Query-specific Subtopic Clustering

Sumanta Kashyapi
University of New Hampshire
Durham, USA
Sumanta.Kashyapi@unh.edu

Laura Dietz
University of New Hampshire
Durham, USA
dietz@cs.unh.edu

ABSTRACT

We propose a Query-Specific Siamese Similarity Metric (QS3M) for query-specific clustering of text documents. Our approach uses fine-tuned BERT embeddings to train 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. QS3M is able to model the fine-grained similarity between text passages about the same broad topic and also generalizes to new unseen queries during evaluation. The empirical evaluation for clustering employs two TREC datasets and a set of academic abstracts from arXiv. When used to obtain query-relevant clusters, QS3M achieves a 12% performance improvement on the TREC datasets over a strong BERT-based reference method and many baselines such as TF-IDF and topic models. A similar improvement is observed for the arXiv dataset suggesting the general applicability of QS3M to different domains. Qualitative evaluation is carried out to gain insight into the strengths and limitations of the model.

CCS CONCEPTS

• **Information systems** → **Specialized information retrieval**; *Digital libraries and archives.*

KEYWORDS

clustering, topic detection, topic model, neural networks, query-specific clustering, Siamese neural networks, similarity metric

ACM Reference Format:
Sumanta Kashyapi and Laura Dietz. 2022. Query-specific Subtopic Clustering. In *The ACM/IEEE Joint Conference on Digital Libraries in 2022 (JCDL '22), June 20–24, 2022, Cologne, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3529372.3530923

1 INTRODUCTION

Users with conscious information needs [30] have a tendency 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 [16]. Explicit clustering is used as a separate post-processing step, either for search result diversification or to present text passages in groups (each representing one subtopic) which is useful for taxonomic browsing [6, 17]. Ideally, the resulting subtopic clusters should be relevant for the user's information need. This scenario is encountered whenever information on a topic is to be retrieved and organized, such as texts obtained from an archive, books in a digital library, or web search results. While the notion of query and document may vary in different domains, we face similar challenges in identifying query-specific subtopics in all of them.

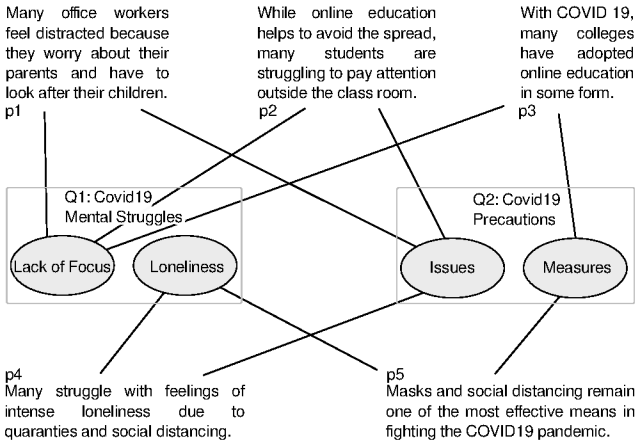

Figure 1: Example of two queries about Covid19 to illustrate the need for query-specific clustering metrics: for query Q1, "Covid19 Mental Struggles", the subtopics "Lack of Focus" and "Loneliness" are more relevant than clusters about "Issues" vs. "Measures"—and vice versa for query Q2. Cluster names are for illustration only and are held-out during training and evaluation.

In this work, we focus on the central step in such an information access system: query-specific subtopic clustering.

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

Researchers have studied subtopic clustering on search results [5, 10, *inter alia*]. A canonical approach to text clustering is to represent passages as vectors which govern a similarity metric (e.g. TFIDF vector with cosine similarity) for hierarchical clustering [19]. Recently, neural embeddings and trained similarity functions obtain better performance [28, 32].

An issue of such clustering approaches is that the similarity score between two passages is the same regardless of the query. This will be the case for any similarity function or embedding scheme that does not take the query into account. We hypothesize that the resulting clustering is more relevant to answer the user's

JCDL '22, June 20–24, 2022, Cologne, Germany

ACM ISBN 978-1-4503-9345-4/22/06…$15.00
https://doi.org/10.1145/3529372.3530923

information need when incorporating the search query into the similarity metric. Even if the same set of passages are relevant for two queries, we may need to cluster them in different ways as depicted in the example in Figure 1.

Hence to produce query-specific subtopic clusters, we design a query-specific text similarity metric, which when used with a distance-based clustering algorithm, will lead to relevant, query-specific clusters of retrieved passages.

Our rationale is that an ideal query-specific similarity metric should identify the query-relevant subtopics and ignore other spurious topical dimensions. We follow a neural approach to train the metric that leverages BERT-based language models. For the given example in Figure 1, it should emit a high similarity score between p_1 and p_3 and low similarity score between passages p_1 and p_4 given the query Q1, "Covid19 Mental Struggles". However, for the other query Q2, "Covid19 Precautions", the scores should change accordingly. Of course, during evaluation, the queries will be different from training queries, suggesting that a query-specific metric learning approach would result in a clustering that is more suitable for the query, and hence more helpful when browsing the retrieved information.

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.

Outline: This paper is organized as follows. Section 2 gives an overview of related work along with the reference method, sentence-BERT [28]. Section 3 elaborates our approach.[1] Section 4 details our empirical evaluation on Wikipedia, while Section 5 evaluates on scientific abstracts from ArXiv. We conclude the paper in Section 7.

2 RELATED WORK

2.1 Text Clustering

Previous research on text clustering [18, *inter alia*] focuses on unsupervised lexical similarity metrics. The similarity metric is used to compute distances between elements in vector space for clustering algorithms [7]. Metzler et al. [22] explore hybrid similarity measures which combine lexical and probabilistic measures with application to query similarity detection. Banea et al. [2] develop an ensemble system that uses a combination of knowledge-based and corpus-based text similarity measures as features. For semi-supervised clustering, researchers have found pairwise binary constraints also known as *"must link"* and *"cannot link"* to be particularly effective [4]. Most lexical similarity metrics employ term-based vector representation of text such as TFIDF. Probabilistic topic models such as latent Dirichlet allocation [8], have been used to extract subtopics from a text corpus and use the topic distribution to represent documents. A natural choice for a similarity metric that uses this representation is the Kullback-Leibler (KL) divergence [29]. Recently, numerous methods are proposed to use deep neural networks in modeling the clustering process. Peng et al. [23] use a reformulation of the traditional k-means algorithm using neural networks. Cho et al. [11] use a modified self-attention mechanism to implement an alternative version of the k-means algorithm.

2.2 Neural Embeddings for Clustering

Clustering algorithms depend on a semantic representation of text. With the advent of transformer-based neural networks [12, 31], text embeddings have given rise to strong linguistic models. It is observed in deep learning and transfer learning research that layers at different levels of a deep network capture specific information about the data [25]. In their work, Peters et al. [24] learned a function that projects the internal state of, ELMo, a deep Bidirectional Language Model which was trained on a large dataset to a contextual embedding space. Being a deep network, BERT [12] also has several layers of attention heads and feed-forward neural networks stacked on top of each other. Researchers have tried to utilize information captured at these layers by averaging all BERT layers [34] or extracting the output of a special token (CLS) in the input [21][26] to obtain a fixed size embedding representing the input sequence. Unfortunately, empirical studies prove that these methods perform poorly in semantic matching tasks. Reimers et al. [28] proposed modifications in retrieving strategies of these embeddings as well as specific fine-tuning techniques that provide better sentence embeddings which perform well in numerous sentence similarity tasks. Next, we provide a brief background on their method known as Sentence-BERT.

Sentence-BERT: After the success of BERT models, researchers made various attempts to use token embeddings obtained from the BERT model to embed longer sequences, such as sentences or documents. Typically, the embedding of the first token embedding (CLS token) from the last layer of a BERT model is used in such scenarios. But instead of relying on a single token and a single layer to capture the embedding of the whole sequence, Reimers et al. explored different ways of combining information from all the layers. They finetune all layers of a BERT encoder using a simple distance metric (e.g. cosine distance or Euclidean distance, etc.) governing the training process. Employing a simpler distance metric in the finetuning process, allowed faster similarity calculation of embedding vectors during inference time.

The key differences between Sentence-BERT and other approaches attempted before it are the following:

1. Averaging hidden layers of the pre-trained BERT model tend to lose vital semantic information captured in separate layers. In contrast, the Sentence-BERT model is fine-tuned based on the outcome of different pooling strategies applied to all the layers of the BERT model.

2. Sentence-BERT utilizes two different network structures to fine-tune BERT embeddings: Siamese networks for pairwise training data (binary similarity regression and multiclass classification) [9] and triplet networks for passage triples (similar and dissimilar passage of a passage referred to as an anchor; together forming a triplet of passages) proposed by Dor et al. [15]

3. Loss functions used to train Sentence-BERT are specialized to tune the embeddings for semantic similarity tasks. a) Regression loss to minimize mean-squared error (MSE) between the similarity labels and cosine similarity between embedding pairs, b) optimization of cross-entropy loss between weighted class probabilities of

[1] Code is available at: https://github.com/nihilistsumo/QS3M. Link to the dataset used for this paper is provided in the github repository.

embedding pairs, c) minimization of triplet loss to reduce the embedding distance between similar passage-pair relative to the same for dissimilar passage-pair in triplets of passages.

2.3 Query-Specific Clustering and Search Result Diversification

Query-specific clustering can be addressed as a separate step, such as the extraction and co-occurrence analysis of keyphrases. Leung et al. [20] develop a query-based clustering method in the context of the user's profile extracted from web snippets of search results. Raiber et al. [27] employ similarity heuristics to perform canopy clustering on the search result set to improve document rankings. Bernardini et al. [5] cluster search results into subtopics using keyphrases. Carpineto et al. [10] and Drosou et al. [16] evaluate the effectiveness of subtopic clustering and search result diversification in post-processing of search results.

The literature suggests that incorporating query information is beneficial for subtopic clustering. However, most of the query-specific subtopic clustering methods rely on matching relevant key phrases from the search results. But keyphrase matching may not be sufficient to capture fine-grained topical information in the search results. Transformer-based embeddings such as Sentence-BERT have been demonstrated to capture high-quality topical information. It is an example of a trained similarity metric, customized for a task, but not specific to a query. Our approach for representing passages is built upon the Sentence-BERT embedding model but incorporates the query while estimating similarity between a pair of representations. Sentence-BERT is used as a strong reference method to empirically demonstrate improvements achieved by our model.

3 APPROACH

Our goal is to design a clustering system for text passages that produces query-relevant subtopic clusters. We focus on training a query-specific similarity metric (henceforth referred to as similarity metric) between text passages, which is used in a distance-based clustering algorithm. This similarity metric should model a topical embedding space that is suitable for identifying query-relevant subtopics. For example, it should emit high similarity scores between passages relevant to the subtopic "Lack of Focus" in the context of the query "Covid19 Mental Struggles" (cf. Figure 1).

Our similarity metric is designed to be used with the following pipeline:

Step 1: Train the similarity metric. We train our model to predict the query-specific similarities: Given a pair of passages, p_1, p_2 and query q, the model predicts whether both passages should share the same subtopic of the training query q.

Step 2: Obtaining pairwise similarity scores. Given a query set Q and retrieved passage sets $\mathcal{P}_q$ for each query $q \in Q$, we apply the model to predict pairwise similarities between all passages in $\mathcal{P}_q$.

Step 3: Clustering based on the similarity metric. 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 is subtopic clusters that coincide with query-specific subtopics. Since there are no convincing solutions for learning the true number of cluster k_q, in this work, we assume knowledge of the true number k_q during evaluation.

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

3.1 Training the Passage Similarity Metric

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 $\mathcal{P}_q$ for a given query q.

We follow a common approach of similarity metric learning where documents are represented in a vector space such that the vector similarity coincides with the semantic similarity of the passages. We focus on neural networks to be able to model such a trainable similarity metric. The open question is how to define a parameterized similarity function that can leverage information about the query.

The novelty of our approach lies in how we model the similarity between passages in a query-specific representation space so that it generalizes to new unseen queries during evaluation time. The similarity is trained end-to-end using training data constructed from a set of queries and a ground truth of ideal passage clusters.

We discuss three neural models in the following, all of which are based on an initial Sentence-BERT representation of query $\vec{q}$ and passages $\vec{p}$ and will predict the similarity score between two passages p_i and p_j. All models are trained end-to-end.

3.1.1 Query-Specific Scaler (QSS):. The first model is based on the assumption that one merely needs to apply the right reweighting of passage embedding representations to arrive at a query-specific similarity. A scaling vector $\vec{q}^{\,\Psi}$ is used for reweighting passage embeddings and is obtained by projecting the query representation $\vec{q}$ with a multi-layer perceptron Ψ. After component-wise reweighting of passage representations,[2] the cosine similarity is used as a measure of the passage similarity.

$$\phi_q(p_i, p_j) = \cos\left(\left(\vec{q}^{\,\Psi} \odot \vec{p}_1\right), \left(\vec{q}^{\,\Psi} \odot \vec{p}_2\right)\right)$$

The projection Ψ (49K trainable parameters) uses a multi-layer perceptron (MLP) with ReLU activation with the same input and output dimensionality (768). The MLP consists of two linear layers with a 32-dimensional "bottleneck" between them, which we found to generalize better than a single larger linear layer.

The QSS model can be interpreted as a Siamese network where the same transformation is applied to a pair of passages through $\vec{q}^{\,\Psi}$ that arises from the query. The model is limited in that the only trainable component Ψ does not have direct access to the passage vectors $\vec{p}$ and is only indirectly trained end-to-end from the loss computed on the resulting similarity.

3.1.2 Query-Specific Siamese Similarity Metric (QS3M):. In the Query-Specific Siamese Similarity Metric we assume that a better similarity metric $\phi_q(p_i, p_j)$ can be obtained with a more complex

[2] Here $\odot$ refers to a component-wise multiplication, where $\cdot$ denotes the dot product.

model to capture the relevance between query and passages. This is captured by a neural network connected in triamese fashion, inspired by the model proposed by Zeghidour et al. [33].

First, vector representations for passages $\vec{p}_i, \vec{p}_j$, and queries $\vec{q}$ are projected into a latent space using the projection function Θ. Next, the projected vectors are integrated into one vector $\vec{z}$ which, in turn, is used to classify the passage pair into "same cluster" vs "different cluster" using a binary classifier Φ. Both Θ and Φ are modeled using multi-layer perceptrons. The goal of the integrated projected vector $\vec{z}$ is to capture the relevance-based similarity between query and passages. We found the most effective combination to be concatenation[3] of passage representations $\vec{p}_i^{\,\Theta}$ and $\vec{p}_j^{\,\Theta}$ with vectors representing the proximity to the query $\left|\vec{p}_i^{\,\Theta} - \vec{q}^{\,\Theta}\right|$ and $\left|\vec{p}_j^{\,\Theta} - \vec{q}^{\,\Theta}\right|$ and the proximity of the passages $\left|\vec{p}_i^{\,\Theta} - \vec{p}_j^{\,\Theta}\right|$.

$$\phi_q(p_i, p_j) = \Phi\left[\vec{p}_i^{\,\Theta}; \vec{p}_j^{\,\Theta}; \left|\vec{p}_i^{\,\Theta} - \vec{q}^{\,\Theta}\right|; \left|\vec{p}_j^{\,\Theta} - \vec{q}^{\,\Theta}\right|; \left|\vec{p}_i^{\,\Theta} - \vec{p}_j^{\,\Theta}\right|\right]$$

We train both Θ and Φ (59K and 38K trainable parameters respectively) across all training queries and passage sets. We model the projection with MLP layers with ReLU activation. Θ consists of two linear layers of size 768 each, while Φ is a single layer MLP of size 5×768. In contrast to QSS, QS3M uses a more expressive embedding space by learning: (1) a shared projection space for queries and passages through Θ, (2) the proximities of projected vectors, and (3) the classification function Φ. The number of additional parameters is kept low by sharing projection parameters for Θ across passages and queries.

3.1.3 Sentence Attention QS3M (attn-QS3M):. So far we used the full paragraph text to generate the passage embeddings. Previous work suggested an advantage of using sentence embeddings [28]. We extend QS3M, by replacing the passage embedding vector Θ with a sentence-attentive passage vector Θ' as follows: Instead of embedding the whole passage at once, individual sentences of the passage are embedded and the sentence vectors are combined using attention mechanism involving the sentences and the query. As an attention mechanism, we use tanh-based additive attention as suggested by Bahdanau et al [1] with 24K trainable parameters.

3.2 Generating Training Data

To train our similarity metric for subtopic extraction, we require a benchmark where for given queries, pairs of passages are labeled with "same cluster" or "different cluster".

Such benchmarks are typically available in two flavors: *flat* clusters benchmarks, where each passage is a member of only one cluster, and *hierarchical* cluster benchmarks, where a parent cluster can be further sub-divided into child clusters. Such query-clustering benchmarks can be derived from a corpus of articles, where each article is associated with a search query, by interpreting each section of the article as one subtopic, as depicted in Figure 2. In this work, we derive a benchmark from Wikipedia articles, but our methods can also be applied to other benchmarks. Because the predominant number of pairs labeled as "different cluster" can negatively impact the training result, we balance the training dataset by sampling

[3]Here [;] denotes vector concatenation, and $|\vec{x}|$ denotes the vector of component-wise absolutes $|x_k|$.

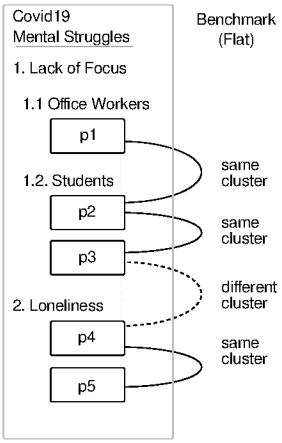

Figure 2: A train/test benchmark for query-specific clustering depicted in Figure 1 can be derived from source articles with sectioned outlines, e.g. Wikipedia articles.

negative pairs. We report results both on balanced and the full (all-pairs) versions of the dataset. Additionally, for *hierarchical clustering* benchmarks, it may be ambiguous whether a pair of passages should be regarded as "same cluster" or "different cluster" when one passage comes from the parent and the other passage from its child cluster. We omit such pairs from our training whenever such ambiguities arise.

4 EVALUATION ON WIKIPEDIA

We use the publicly available Wikipedia article collection from the TREC Complex Answer Retrieval (CAR) [14]. We are using CAR dataset version 2.3 of CAR year 1 for training and evaluation.[4]

The official CAR task is a passage ranking task where subtopics are given as queries. While no clustering task was offered, we use CAR's training/testing articles to derive a benchmark for our work as depicted in Figure 2. We follow CAR's definition of queries as article titles. We use paragraphs in each article as an ideal set of retrieved passages $\mathcal{P}_q$ to be clustered leaving integrated ranking and clustering tasks for future work. We derive hierarchical and flat clustering benchmarks as described in Section 3.2, where each section is interpreted as one subtopic, i.e., one ground truth gold cluster of passages. While this benchmark is automatically generated, it has been demonstrated to align well with relevance judgments of human assessors [13].

Datasets: We use the following datasets for training, pre-training, and evaluation from CAR year 1. The CAR collection is based on Wikipedia articles, where administrative headings such as "References", and "See also" are filtered out, articles with less than three sections are removed and, articles from general categories are preferred.[5] All datasets have separate, non-overlapping query sets, to avoid test data leakage into the training process. Dataset statistics are given in Table 1.

[4]TREC CAR data set: http://trec-car.cs.unh.edu/

[5]CAR data cleaning: trec-car.cs.unh.edu/process/dataselection.html

Table 1: Dataset statistics of flat and hierarchical benchmarks. The last columns provide the average number of clusters per query and the average number of passages per cluster, with standard deviation.

Dataset	Psg. pairs	Queries	Clusters per Query		Passages per Cluster	
			Flat	Hierarchical	Flat	Hierarchical
CAR-A	168K	125	8 ±2.27	17 ±10.49	6 ±7.23	3 ±2.10
CAR-B	118K	115	7 ±2.05	16 ±10.59	6 ±7.61	2 ±2.14

Pre-training: Using 1.6 million queries in `train.v2.0`[6] (after omitting queries in CAR-A and CAR-B), we generate data to pretrain Sentence-BERT [28] (referred as SBERT hereafter) with a maximum input sequence length of 512.

Training: Analogously, we create training data for the similarity metric using 162,000 queries from the rest of `train.v2.0`. To avoid overfitting to particular topics, we choose three paragraphs for each query, two from the same subtopic, and one from a different subtopic.

CAR-A: For evaluation, articles in `benchmarkY1test` are converted to flat and hierarchical clustering benchmarks as described in Section 3.2.

CAR-B: Analogous to CAR-A, but using the `benchmarkY1train` dataset. Despite its name, these queries are held out from our training and are used only for evaluation.

4.1 Conducted Experiments

We evaluate the steps 2 and 3 of our query-specific subtopic clustering pipeline using the following experiments:

Experiment 1 (Similarity Metric): We evaluate how well the trained passage similarity metric generalizes to new queries. We use the macro-averaged area under the Receiver Operating Characteristic curve (AUC) as our evaluation metric (higher is better). We evaluate both the balanced (akin to the training data used during training) and the full benchmark consisting of all passage pairs.

Experiment 2 (Clustering): We evaluate the quality of the clusters obtained by the trained similarity metric using an average-link hierarchical agglomerative clustering algorithm. The generated clusters are evaluated in terms of the macro-averaged Adjusted RAND index (ARI), a clustering metric reflecting the degree of agreement between the clustering ground truth and the obtained clusters. ARI is adjusted for to account for the prevalence of negative pairs in ground truth and predictions.

4.2 Compared Variations and Baselines

We evaluate the following query representations:

Title: Embedding of a short keyword query (aka title query). CAR uses the article title in lieu of a web search query. Section headings are excluded.

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

Description: Embedding of a longer query description. Here we use the text above the first heading (which is omitted from the passage set $\mathcal{P}_q$).

Passages: The average of embeddings across all passages in the query's passage set $\mathcal{P}_q$ (akin to ideas of pseudo-relevance feedback).

We train three variations of our similarity metric: QSS, QS3M, and sentence-attentive QS3M (attn-QS3M). We use a pre-trained SBERT to derive initial representations of queries $\vec{q}$ and passages $\vec{p}$. We also experiment with raw BERT embeddings without the SBERT pretraining step but observe that this degrades performance, hence we only report the best variant: QS3M-rawBERT-Description.

As a strong baseline, we use SBERT [28], a recent BERT-based [12] reference method, re-trained on our data (see pretraining). Following are the baselines included in our evaluation:

SBERT euclid: Euclidean distance of SBERT embedded passages [28].
SBERT cosine: Like SBERT euclid, but using the cosine similarity.
Jaccard: Set-based similarity between sets of words from passages.
TFIDF: Cosine similarity between TFIDF vectors of passage words.
Topic model: Jensen-Shannon divergence between the topic distribution of two passages, estimated using an LDA topic model with 200 topics [8]. The topic model is trained on our training set.

4.3 Experimental Results

4.3.1 Experiment 1 (Similarity Metric): We study to which extent the trained similarity metrics are able to generalize to new, unseen queries in *CAR-A* and *CAR-B*, using both the flat and hierarchical clustering benchmark.

Table 2 presents the empirical results for classifying passage pairs into the same vs different clusters as measured in ROC-AUC. We observe that our methods QSS and QS3M perform significantly better than all baselines, specifically SBERT cosine and SBERT euclid. While QS3M, QSS, and SBERT, are using the same underlying BERT-based representation, the difference is that only QS3M and QSS train a query-specific similarity metric. These results demonstrate that incorporating the query into the similarity metric improves its prediction quality. However, we also observe that attn-QS3M performs worse than the baselines. This indicates that embedding passages as a whole are more suitable than the weighted sum of individual sentence embeddings. We speculate that individual sentences are often lacking the necessary context to obtain a meaningful representation.

4.3.2 Experiment 2 (Clustering): We evaluate to which extent the improvements in the similarity metric give rise to better clustering results. We use average-link hierarchical agglomerative clustering to obtain subtopics as clusters of passages for each query. We use the macro-averaged Adjusted RAND index as a measure of clustering quality for both *Flat* and *Hierarchical* benchmarks. The evaluation results are reported in Table 2.

We observe that similarity metrics with better pairwise performance (Table 2) also lead to better clustering performance. In particular, QS3M is the best performing method, achieving on average 12% relative improvement over the best-performing baseline method. For both CAR-A and CAR-B, QS3M achieves statistically

Table 2: Left: Experiment 1. Evaluation of the similarity metric for predicting same-vs-different cluster on the flat benchmark. Performance is measured in terms of macro-averaged AUC. Right: Experiment 2. Clustering performance of agglomerative clustering with the similarity metric, measured in macro-averaged Adjusted RAND index (ARI). Both: Baseline methods are at the bottom. Significantly higher ▲ or lower ▼ method according to a paired t-test ($\alpha = 0.05$, after Bonferroni correction $\alpha = 0.003$) with respect to SBERT euclid (marked with ⋆), which is the best performing baseline.

	Exp 1: Same-vs-different cluster (AUC)				Exp 2: Cluster Quality (ARI)			
	CAR-A		CAR-B		Flat		Hierarchical	
Methods	balanced	all-pairs	balanced	all-pairs	CAR-A	CAR-B	CAR-A	CAR-B
QS3M-Passages	0.744	0.734	0.760	0.749	**0.300**▲	0.307	**0.237**	**0.276**
QS3M-Description	**0.751**▲	**0.745**▲	**0.764**	**0.759**▲	0.298▲	**0.323**	0.233	0.274
QS3M-Title	0.750▲	0.738▲	0.763	0.751▲	0.289▲	0.306	0.217	0.246
QSS-Passages	0.737	0.726	0.745	0.738	0.249	0.295	0.219	0.226
QSS-Description	0.746▲	0.735	0.761	0.748▲	0.263	0.304	0.221	0.255
QSS-Title	0.745	0.732	0.758	0.745	0.269	0.296	0.225	0.239
attn-QS3M-Passages	0.668▼	0.661▼	0.676▼	0.669▼	0.178▼	0.183▼	0.142▼	0.154▼
attn-QS3M-Description	0.681▼	0.674▼	0.678▼	0.681▼	0.195▼	0.215▼	0.164▼	0.178▼
attn-QS3M-Title	0.694▼	0.682▼	0.697▼	0.687▼	0.211▼	0.220▼	0.160▼	0.190▼
QS3M-no-query	0.747	0.734	0.761	0.747	0.284	0.297	0.218	0.241
QS3M-rawBERT-Description	0.727▼	0.715▼	0.738▼	0.726▼	0.232▼	0.254▼	0.199▼	0.207▼
SBERT euclid [28]⋆	0.741⋆	0.730⋆	0.746⋆	0.739⋆	0.263⋆	0.295⋆	0.214⋆	0.239⋆
SBERT cosine	0.740	0.728	0.745	0.737	0.258	0.287	0.216	0.236
Jaccard	0.615▼	0.600▼	0.622▼	0.603▼	0.069▼	0.066▼	0.109▼	0.124▼
TFIDF cosine	0.618▼	0.597▼	0.624▼	0.591▼	0.071▼	0.068▼	0.109▼	0.120▼
Topic Model [29]	0.464▼	0.474▼	0.456▼	0.475▼	≈ 0▼	0.009▼	≈ 0▼	≈ 0▼

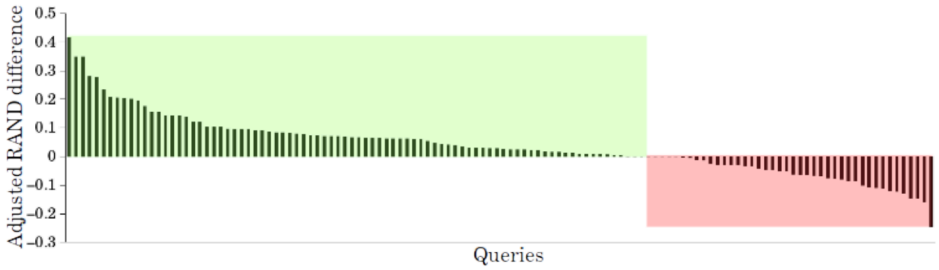

Figure 3: Helps-Hurts analysis: QS3M-meanall (top) vs. SBERT (bottom) on CAR-A Flat. Each column corresponds to a particular query and the height value corresponds to the difference between QS3M-meanall and SBERT euclid in terms of the ARI score obtained for that query. QS3M-meanall performs better than SBERT for around 70% of the queries.

significant improvements with respect to both clustering benchmarks. In contrast, the simpler QSS model does not achieve as good results.

We observe a large variance in cluster quality across queries. Hence, we perform a helps-hurts analysis to compare the clustering performance of the best QS3M model with the best SBERT baseline on a per-query basis. As displayed in Figure 3, for two-thirds of queries in *CAR-A Flat*, QS3M received a better adjusted RAND index than SBERT.

4.3.3 Best query representation: We explore three different query representations for Experiments 1 and 2 (Table 2). We observe that the query representations *description* and *passages* achieve better results than *title*. We believe that the issue arises because the query titles only contain a few keywords which are not sufficient to capture useful context information. In contrast, QS3M without any query representation (QS3M-no-query) is worse than any other QS3M variant, while performing on par or better than the SBERT baseline. This suggests that in order to obtain relevant subtopics it is instrumental to train a query-specific similarity metric.

4.3.4 Hierarchical clustering: In Table 2 we observe that results for hierarchical clustering are consistently lower than results for flat clustering, despite being derived from the same dataset. The reason is that the hierarchical dataset has more clusters than the flat dataset, for which we offer the following explanations: While top-level sections on Wikipedia are discussing distinct subtopics, while other the topics of subsections only have subtle differences. Furthermore,

Table 3: Categories used in our arXiv experiment.

ArXiv category (Subject part)	Category title (used as query)	Subtopics	Abstracts per subtopic (stdev)
eess	Electrical Engineering and Systems	3	167 (158.6)
astro-ph	Astrophysics	6	83 (32.6)
cond-mat	Condensed Matter	9	55 (35.0)
nlin	Nonlinear Systems	5	100 (68.1)
q-bio	Quantitative Biology	10	50 (43.4)
q-fin	Quantitative Finance	9	55 (20.2)
stat	Statistics	5	100 (86.4)

since the difficulty of this task increases with the number of true clusters. Furthermore, many hierarchical gold clusters have only three or fewer passages (cf. Table 1), rendering this a challenging data set for agglomerative clustering.

5 EVALUATION ON ARXIV

In this section we demonstrate that our method can be easily generalized for a different domain, such as academic publication abstracts from arXiv.[7] ArXiv provides of abstracts of scientific papers with information along with their respective field of study referred in the dataset as "categories". We leverage the taxonomy of arXiv categories[8] to identify the query and the subtopic corresponding to each abstract. For example, if an abstract is annotated econ.EM as the arXiv category, then we use the broader category or the subject area "Economics" (econ) as the query and "Econometrics" (EM) as the subtopic under "Economics".

We address the task of subtopic clustering under each subject area as a separate query. Given a name of the subject as query q and a set of abstracts $\mathcal{P}_q$ associated with the subject, cluster these abstracts so that each cluster represents exactly one subtopic.

Our experiments are based on a subset of the full dataset, with categories listed in Table 3 along with other statistics. We randomly sample 500 paper abstracts for each subject to construct the dataset. We divide the subset into two folds for 2-fold cross-validation, training on one fold and evaluating on the other and vice versa. While generating the folds, we maintain the per-subtopic distribution of abstracts for each fold. Note that, we do not use any parts of the Wikipedia dataset from the previous experiments for training to clearly observe the effect of adapting to the arXiv domain. The dataset along with the code for this experiment will be provided as part of the online appendix.

We evaluate the performance of the QS3M-Title variant of our method and compare it with one of our baseline approach SBERT-cosine both in terms of pairwise accuracy (AUC) and clustering accuracy (adjusted RAND Index).

Experimental results in terms of pairwise classification accuracy (AUC) and clustering performance (ARI) are presented in Table 4 and Figure 4 respectively. We observe that the baseline approach SBERT-cosine performs poorly on this dataset (a random baseline would obtain an AUC of 0.5). In contrast, our method QS3M-Title obtains an AUC of more than 0.7, which indicates moderate agreement with the gold standard.

[7]https://www.kaggle.com/Cornell-University/arxiv – As downloaded in September, 2021.
[8]https://arxiv.org/category_taxonomy

Table 4: Experiments on the arXiv dataset.

Methods	Fold 1 AUC	Fold 2 AUC
QS3M-Title	0.773	0.734
SBERT cosine	0.592	0.598

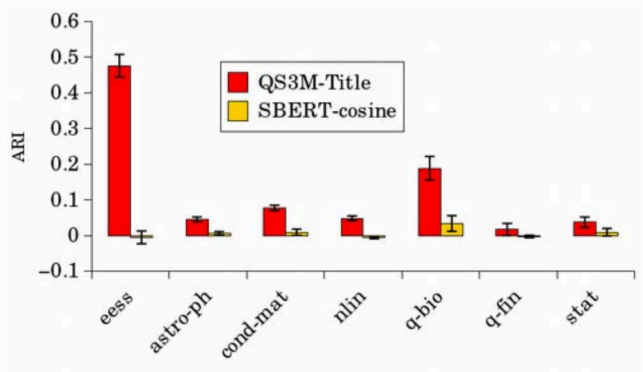

Figure 4: Per-query clustering evaluation results on arXiv dataset. The plot reports average clustering evaluation across the two folds in terms of adjusted RAND index (ARI) with standard error bars. ARI can range between -1 and +1.

We present the clustering results for each query q in both folds in Figure 4. We observe that our method achieves large improvements over the SBERT baseline, suggesting that our method finds subtopics more reliably. In particular, our method works well for the subject `eess` (Electrical Engineering and Systems Science) and `q-bio` (Quantitative Biology). This suggests that both of them have distinctive subtopics with little topical overlap. Also, documents under different subjects are very different (e.g. scientific papers on Astrophysics and Quantitative biology are expected to be very different from each other) when compared to that of the CAR dataset. Hence, a query-specific similarity metric will be more suitable for this dataset than a general similarity metric that does not depend on the query.

Overall, these results suggest that our query-specific clustering approach can be easily transferred to other domains with relatively little training.

6 QUALITATIVE EVALUATION

We conduct a qualitative analysis of the clustering evaluation. The aim of this analysis is to gain insight into the strengths of our proposed model with the help of some examples from the dataset.

Consider the following pair of passages from **CAR-A** dataset on the query *"Gardening"*:

> **Passage 1:** *Gardens in Renaissance were adorned with sculptures, topiary and fountains. In the 17th century, knot gardens became popular along with the hedge mazes. By this time, Europeans started planting new flowers such as tulips, marigolds and sunflowers.*

> **Passage 2:** *Islamic gardens were built after the model of Persian gardens and they were usually enclosed by walls and divided in 4 by watercourses. Commonly, the center of the garden would have a pool or pavilion. Specific to the Islamic gardens are the mosaics and glazed tiles used to decorate the rills and fountains that were built in these gardens.*

From only the text of the passages, it is quite difficult to decide whether they are similar enough to share the same subtopic cluster. In the context of human culture, they are topically distinct but in the context of the history of gardening, they are similar. Due to this ambiguity, SBERT-cosine, which does not have access to the query, assigned a low similarity score to this passage pair. In the CAR benchmark, both passages are relevant for the query "Gardening" in the gold cluster "History". By taking into account that the information need is about gardening, our QS3M model correctly identifies a high similarity between these passages.

The previous example is a case of false-negative which is rectified by the QS3M approach. However, we find this to be a rare instance across both the CAR-A and CAR-B datasets. The main reason why our QS3M approach achieves better results is its ability to avoid false-positive cases. Let us consider another pair of passages from the CAR-A dataset on the query:

> **Passage 1:** *Big-game fishing started as a sport after the invention of the motorized boat. In 1898, Dr. Charles Frederick Holder, a marine biologist and early conservationist, pioneered this sport and went on to publish many articles and books on the subject noted for their combination of accurate scientific detail with exciting narratives.*
>
> **Passage 2:** *In addition to capturing fish for food, recreational anglers might also keep a log of fish caught, either in a physical form or with technology such as the FISHBUOY or Fishbrain mobile logging application, and submit trophy-sized fish to independent record keeping bodies. In the Republic of Ireland, the Irish Specimen Fish Committee [...]. It also uses a set of 'fair play' regulations to ensure fish are caught in accordance with accepted angling norms.*

Without knowledge of the query, these two passages could share topics such as *"fishing"* and *"sport"* and consequently influence non-query specific methods such as SBERT-cosine to incorrectly assign high similarity scores. However, knowing that these passages are retrieved for the query *"Recreational fishing"*, it becomes apparent that they belong to two different clusters, *"History"* and *"Fish logs"*, as correctly identified by *QS3M*.

Error Modes: From Figure 3, we observe that some queries did not benefit from the QS3M model. To investigate why our model failed to improve upon SBERT for those queries, we analyzed the queries and their content in detail. Interestingly, most of these worst performing queries are related to food or nutrition, such as *Bagel, Christmas pudding, or Fudge*, which points towards a broader context. Naturally, all of these queries share similar subtopics such as recipes, different varieties, and history. Hence, a simple template-learning model, e.g. as proposed by Banerjee et al. [3], would solve the clustering problem for all of these similar queries. However, the focus of our study is to solve the more difficult problem where results for queries would not be appropriately represented by a fixed template outline. Indeed, most of the queries in the CAR dataset and the arXiv dataset (each category corresponds to a different area of study) are of this type and benefit greatly from our QS3M model.

7 CONCLUSION

In this work, we propose a query-specific similarity metric, suitable for query-relevant subtopic clustering of passages. Traditionally, the query only indirectly influences the clustering result through the candidate set generation. We propose a more direct approach toward query-specific clustering and demonstrate that clustering results can be improved by 12% with our Query-Specific Siamese Similarity Metric (QS3M). QS3M is trained to decide if two passages should be placed in the same versus different subtopics for a given query. Our method utilizes BERT-based representations of passage and query content to machine-learn a query-specific projection of passages into a similarity space. Our approach is different from task-specific metric learning in that test queries are not known at training time. We demonstrate the improvement using two TREC datasets and one arXiv dataset on both flat and hierarchical query-specific clustering benchmarks. On all test sets, QS3M outperforms a strong, BERT-based reference method of Reimers et al. [28], our simpler variant QSS, and many other baselines including TF-IDF and topic models.

While topic models are appealing as they do not require training data, in Table 2 we demonstrate that they are not able to identify fine-grained topics such as article sections. While our method is supervised, we demonstrate that suitable training data can be readily derived from Wikipedia (cf. Section 3.2), sufficient to generalize to unseen queries and new subtopics.

Query-specific clustering can be applied to any context-specific text clustering task, such as detecting subtopics in corpora, domain-specific taxonomy extraction, faceted information access, and search diversification. It can be used to identify topical dimensions of a conversational search dialog, trending subtopics on Twitter, as well as to identify sections for automatic article generation. As our similarity metric relies on latent representations of passages, it can even be applied to multilingual settings as long as suitable embedding models exist for these languages. The generalizability of our approach for other clustering algorithms (e.g. k-means) remains to be explored.

ACKNOWLEDGMENTS

We thank Dr. Federico Nanni for his helpful discussion regarding the analysis of the results. We also thank the anonymous reviewers for their constructive feedback.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] 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*. Association for Computational Linguistics, 635–642.

[3] Siddhartha Banerjee and Prasenjit Mitra. 2015. Wikikreator: Improving wikipedia stubs automatically. 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)*. 867–877.

[4] 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.

[5] 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*, Vol. 1. IEEE, 206–213.

[6] Michael S Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam, and Ed H Chi. 2010. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 303–312.

[7] 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*. ACM, 11.

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

[9] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems* 6 (1993).

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

[11] Minsik Cho, Keivan A Vahid, Saurabh Adya, and Mohammad Rastegari. 2021. DKM: Differentiable K-Means Clustering Layer for Neural Network Compression. *arXiv preprint arXiv:2108.12659* (2021).

[12] 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* (2018).

[13] Laura Dietz and Jeff Dalton. 2020. Humans Optional? Automatic Large-Scale Test Collections for Entity, Passage, and Entity-Passage Retrieval. *Datenbank-Spektrum* (2020), 1–12.

[14] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview.. In *TREC*.

[15] Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. 2018. Learning thematic similarity metric from article sections using triplet networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 49–54.

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

[17] Susan Dumais, Edward Cutrell, and Hao Chen. 2001. Optimizing search by showing results in context. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 277–284.

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

[19] 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*, Vol. 4. 9–56.

[20] 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 (2008), 1505–1518.

[21] Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561* (2019).

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

[23] Xi Peng, Ivor W Tsang, Joey Tianyi Zhou, and Hongyuan Zhu. 2018. k-meansnet: When k-means meets differentiable programming. *arXiv preprint arXiv:1808.07292* (2018).

[24] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*. 2227–2237.

[25] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1499–1509. https://doi.org/10.18653/v1/D18-1179

[26] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531* (2019).

[27] 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*. 333–342.

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[29] Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* 427, 7 (2007), 424–440.

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

[31] 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*. 5998–6008.

[32] 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*. 62–69.

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

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