



# Research proposal content extraction using natural language processing and semi-supervised clustering: A demonstration and comparative analysis

Benjamin M. Knisely<sup>1</sup> · Holly H. Pavliscsak<sup>1</sup>

Received: 9 December 2022 / Accepted: 7 March 2023 / Published online: 8 April 2023  
© Akadémiai Kiadó, Budapest, Hungary 2023

## Abstract

Funding institutions often solicit text-based research proposals to evaluate potential recipients. Leveraging the information contained in these documents could help institutions understand the supply of research within their domain. In this work, an end-to-end methodology for semi-supervised document clustering is introduced to partially automate classification of research proposals based on thematic areas of interest. The methodology consists of three stages: (1) manual annotation of a document sample; (2) semi-supervised clustering of documents; (3) evaluation of cluster results using quantitative metrics and qualitative ratings (coherence, relevance, distinctiveness) by experts. The methodology is described in detail to encourage replication and is demonstrated on a real-world data set. This demonstration sought to categorize proposals submitted to the US Army Telemedicine and Advanced Technology Research Center (TATRC) related to technological innovations in military medicine. A comparative analysis of method features was performed, including unsupervised vs. semi-supervised clustering, several document vectorization techniques, and several cluster result selection strategies. Outcomes suggest that pretrained Bidirectional Encoder Representations from Transformers (BERT) embeddings were better suited for the task than older text embedding techniques. When comparing expert ratings between algorithms, semi-supervised clustering produced coherence ratings ~25% better on average compared to standard unsupervised clustering with negligible differences in cluster distinctiveness. Last, it was shown that a cluster result selection strategy that balances internal and external validity produced ideal results. With further refinement, this methodological framework shows promise as a useful analytical tool for institutions to unlock hidden insights from untapped archives and similar administrative document repositories.

**Keywords** Text mining · Machine learning · Cluster validation · Document clustering · Research portfolio

---

**Disclaimer:** The views, opinions and/or findings contained in this publication are those of the authors and do not necessarily reflect the views of the Department of Defense and should not be construed as an official DoD/Army position, policy or decision unless so designated by other documentation. No official endorsement should be made. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government.

---

Extended author information available on the last page of the article

## Introduction

The goal of funding scientific research is typically to benefit some societal want or need. In economic terms, scientific research provides the supply of knowledge to satisfy these wants and needs, and these societal wants and needs generate the demand for this knowledge (Sarewitz & Pielke, 2007). Funding institutions and policy-makers serve as the interface between these two elements by soliciting proposals for research when market forces have not “naturally” prompted private sector entities to address these research needs. Balancing the relationship between supply of research and demand by society through research solicitation is a significant challenge for policy-makers (Edler & Boon, 2018; McNie, 2007).

Millions of proposals are submitted to funding institutions every year across a plethora of fields. Many of these proposals are accepted, however, most of these proposals are rejected and archived in various universities and institutions (Boyack et al., 2018). The information contained in these proposals could be used to identify trends in research related to a specific domain. This data can also reveal gaps in the supply of research proposals and demand initiated by institutions in response to societal values and needs. Institutions could use this information to guide allocation of funds and the generation of calls for research, and researchers could use this information to channel their efforts.

## Document clustering

Summarizing trends across collections of research documents requires intense manual review that can be time prohibitive and quite subjective. Clustering algorithms can relieve some of the burden of the review task and can eliminate some reviewer-induced bias. Clustering is an unsupervised machine learning technique that algorithmically partitions unlabeled data into groups, or clusters, based on some measure of similarity. Popular clustering algorithms include *k*-means clustering, hierarchical (or agglomerative) clustering, gaussian mixture-models, and density-based algorithms like DBscan.

Document clustering uses algorithms to partition collections of documents into groups with semantically similar information to make analysis of documents more manageable (Subakti et al., 2022). Document clustering has been applied to text in many contexts, for example social media (Curiskis et al., 2020), medicine (Sandhiya & Sundarambal, 2019), law (Bhattacharya et al., 2022; Dhanani et al., 2021), hospitality (Kaya et al., 2022), patents (Choi & Jun, 2014; Kim et al., 2020), regulatory data (Levine et al., 2022), and engineering documents (Arnarsson et al., 2021). There are many unique challenges associated with document clustering. Text data is often noisy and must be carefully preprocessed using natural language processing (NLP) techniques (Bird et al., 2009). Further, text must be translated to a numerical vector representation prior to clustering. Common approaches to convert text to vector representation include bag-of-words methods and TF-IDF (term frequency-inverse document frequency), word embedding models such as word2vec (Mikolov et al., 2013) and Global Vectors for Word Representation (GloVe) (Pennington et al., 2014), and transformer models like Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). Summarizing and interpreting output document clusters can also be difficult due to the high-dimensionality of text-based data and is an active area of research (Afzali & Kumar, 2019; Penta & Pal, 2021).

## Clustering research documents

Quantitative analysis of text-based scientific material for the purpose of text summarization, specifically clustering, has received much attention in the fields of information management and informetrics (Ebadi et al., 2020; Jiménez et al., 2021; Mishra et al., 2022; Zhang et al., 2018; Zhou et al., 2019). Much of this work has focused on analyzing scientific research documents, as opposed to proposal documents (Boyack et al., 2018). These two types of documents represent supply of research at two different stages of the supply chain and are both worthy of analysis for the purpose of modeling scientific evolution.

The authors were able to locate relatively few recent articles specifically addressing research proposal clustering (Freyman et al., 2016; Ma et al., 2012; Priya & Karthikeyan, 2014; Rajput & Kandoi, 2017; Saravanan & Babu, 2021; Wang et al., 2015). The most common motivation in these papers was to systematically streamline proposal assignment to reviewers based on discipline. Other recent research sought to cluster funded research proposals to map disciplines within research portfolios (Nichols, 2014; Talley et al., 2011). In both cases, cluster structures were optimized to maximize homogeneity of discipline-related content within clusters. Clusters generated for this purpose may not meet the analytical needs of an institutions trying to summarize historic proposal content for other analytical purposes. Institutions therefore require a method to cluster proposals where cluster generation is tailored to capture specific themes of interest. Semi-supervised clustering could enable this capability.

## Semi-supervised clustering

In contrast to traditional unsupervised clustering, semi-supervised clustering uses prior knowledge regarding the structure of the data to enhance the performance of the clustering algorithm. Most semi-supervised clustering algorithms require a sample of the dataset to be class-labeled. These classes are then used to define “Must-Link” (ML) and “Cannot-Link” (CL) relationships, where datapoints in the same class can be defined as ML and datapoints not in same the class as CL. Other algorithms leverage information regarding outcome variables associated with suspected clusters in the data, for example by minimizing the prediction error associated with an outcome variable given generated clusters (Ghahemi et al., 2022). Semi-supervised document clustering is also a small but active area of research. Sadjadi et al. (2021) described and demonstrated a “concept-based” semi-supervised clustering process that leverages a cluster-purity system, where generated clusters that contain labeled data are segmented based on several rules. Mei (2019) proposed a new type of supervising information (“subset partitions”) and demonstrated the algorithm with a document clustering task. Hu et al. (2016) introduced the concept of feature supervision, where the analyst labels features (words) of documents that have discriminating power in addition to labeling the entire document for constraint- or seed-based semi-supervised learning.

## Research objectives and contributions

There is currently a lack of methods and guidance to support research proposal clustering for targeted thematic analyses. There are two main objectives of this research. First, this work establishes an end-to-end methodology to partially automate categorization of proposal document archives based on thematic areas of interest using semi-supervised

clustering. This method takes advantage of both structured insights from subject matter experts as well as machine learning to produce proposal categories. Results of this research include guidance and best-practices for reproduction by practitioners and a framework for continued validation by other researchers.

The second objective of this research was to provide a comparative analysis of methodological features to support optimal application. This includes:

1. Evaluating the performance of the method using several state-of-the-art baseline approaches for document vectorization.
2. Evaluating the performance of the method with semi-supervised clustering compared to unsupervised clustering.
3. Evaluating several strategies for selection of cluster result candidates based on quantitative internal and external validation metrics.

A case study was devised to demonstrate the method and facilitate analysis. This objective was to cluster proposals submitted to the US Army Telemedicine and Advanced Technology Research Center (TATRC) Advanced Medical Technology Initiative (AMTI). AMTI seeks to identify and demonstrate key emerging technologies related to military medicine and provides short-term funding opportunities that support this goal. AMTI was specifically interested in categorizing proposals based on the key problems the applicants proposed to address.

## Methodology

This section describes a generalized methodology that can be applied to generate annotations for any proposal database using semi-supervised clustering. Figure 1 provides an overview of this method. In Sect. “[Demonstration of method](#)” the method is demonstrated on a specific application.

### Manual document annotation

Semi-supervised clustering requires partially labeled datasets. At least two domain experts are suggested to manually annotate proposals to help establish reliability. How the documents are annotated depends on the goals of the analyst. Analysts should tailor their annotations corresponding to the insights they seek to gain. For example, one might be interested in identifying trends in the problems being addressed in proposals, or the technological solutions they propose. This annotation strategy will guide cluster boundary definition in next steps.

Qualitative coding is a systematic process for assigning labels to bodies of text (Castleberry & Nolen, 2018). The coding process used in this work is as follows: First, domain experts establish criteria and goals for labeling the proposals. Second, each expert should individually read the documents and inductively generate thematic codes (Kalpokaite & Radivojevic, 2019). Third, the experts should convene and compare codes. There should be open discussion to reconcile differences, and a final list of codes should be agreed upon. Fourth, each expert should return to the documents and manually apply the final list of codes. These codes can then be compared to identify disagreement. Percent agreement or inter-rater reliability can be measured at this stage

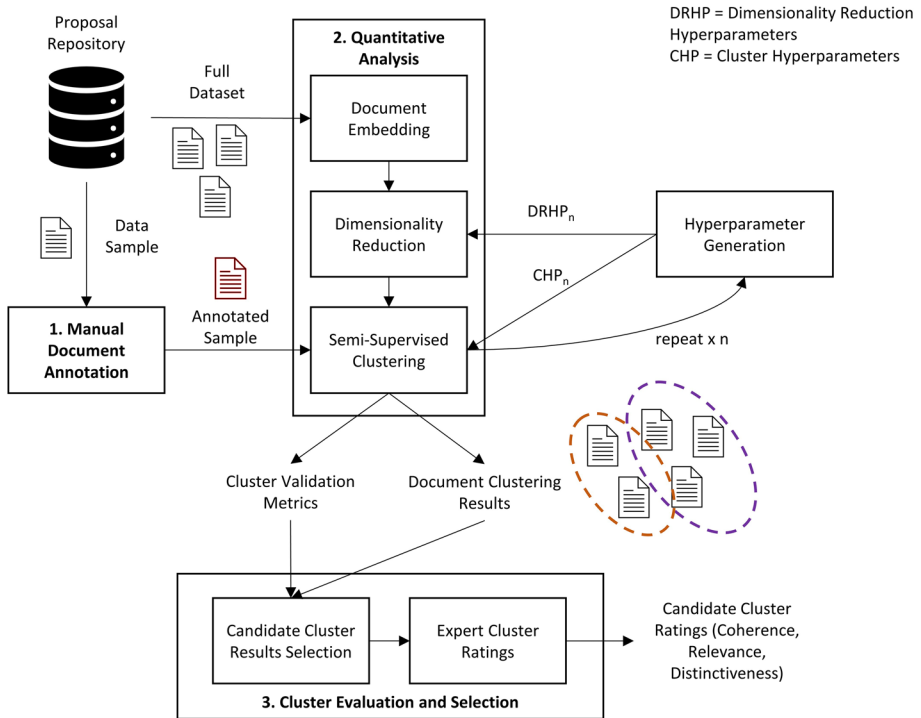


Fig. 1 Overview of methodology

to evaluate the reliability of the coding process (Bajpai et al., 2015). Last, differences in code application should be discussed and reconciled with a final rule by vote or mutual agreement.

There may be instances where a proposal does not fit well into a single category. There are three main ways to address this: (1) make a call—pick the category that is the most relevant to the proposal; (2) merge categories—if there is significant overlap between two or more categories, consider whether these need to be separate categories at all; (3) new category—if there is significant overlap between two or more categories, and it is preferred not to merge them, then create a new category at the intersection of those categories.

There is little existing guidance on how many documents should be annotated. While the exact proportion of documents that must be labeled is likely dataset specific, as little as 5%–10% of labeled data may enhance model performance (Zhong, 2006). More work is required to provide additional guidance on this topic. From a practical perspective, the number of annotated documents should correspond to the amount time and resources available.

### Quantitative analysis

This section discusses the clustering and quantitative evaluation process.

## Document embedding

Text data must be translated into a vector format, or an embedding, prior to use as input to clustering (Almeida & Xexéo, 2019). Vectorizing text allows algebraic operations to be performed on text-based data. Traditional means for deriving text embeddings were constructed on a document-level and were based on the frequency, importance, and co-occurrence of words. Popular implementations include bag-of-words methods and TF-IDF (term frequency-inverse document frequency). These methods often result in large, sparse matrices ( $\# \text{ unique words/tokens} * \# \text{ documents}$ ) that can be burden memory. Further, these representations treat every word as a unique feature, and therefore do not capture semantic similarity between words.

Recent efforts in document clustering have taken advantage of pre-trained text embeddings, a form of transfer learning, which are dense vector representations of text on a word- or sentence-level derived using statistical models trained on massive corpuses of text (Curiskis et al., 2020; Y. Li et al., 2020; Mohammed et al., 2020). By training on massive corpuses of text, these embeddings are able to represent semantic meaning embedded in the text in a latent vector space (Khattak et al., 2019; Liu et al., 2020). Word and sentence embeddings can be subjected to algebraic operations such that semantic meaning is preserved, allowing similarity between bodies of text to be quantified. These embeddings fall into two categories, static and contextualized. Static word embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), produce a fixed vector for individual words regardless of context. Contextualized embeddings, such as BERT (Devlin et al., 2019) produce vector embeddings that capture information regarding the word in isolation as well as the context of the word globally.

**Preprocessing text for embeddings** Appropriate preprocessing depends on the intended embedding technique. For bag-of-word-style embeddings and static word embeddings, the following process can be used. First, text is tokenized. This is a process where individual words or other text features are demarcated into distinct elements. Text should then be lower-cased such that identical words with different cases are recognized as identical. Next, it is common to remove punctuation and stop words. Stop words are common words that provide little information, such as “is”, “the”, “that”, “there”, “a”, and “are”. Last, tokens should be shortened to their base form. This is often accomplished with lemmatization. Lemmatizers are models that transforms words into their base morphological form. Examples transformations include changing “children” to “child”, or “running” to “run”. Many natural language processing packages, such as python’s natural language toolkit (NLTK), have prebuilt functions to perform these tasks (Bird et al., 2009).

In contrast, for transformer-based embeddings such as BERT, these preprocessing tasks may not be necessary at all. The BERT documentation (<https://huggingface.co/docs/transformers/preprocessing>) suggests that for pretrained embeddings, documents only need to be tokenized and truncated to the maximum required length. BERT models have a maximum length of token sequences (e.g., 512) that they can handle and therefore must be truncated (Pappagari et al., 2019). Truncation strategies for transformer models are mixed and is an open area of research (Mutasodirin & Prasojo, 2021; Sun et al., 2019).

## Dimensionality reduction and semi-supervised clustering algorithms

For high-dimensional data, it is common to apply dimensionality reduction techniques prior to clustering to manage the “curse of dimensionality” which can promote over-fittings and hinder algorithm performance (Mittal et al., 2019; Molchanov & Linsen, 2018). These techniques attempt to condense the information contained in the features of a dataset to a smaller number of latent dimensions. There are several popular dimensionality reduction techniques, such as principal component analysis (PCA) and linear discriminant analysis (LDA) (Reddy et al., 2020). These are relatively simple algorithms that are easy to implement but are limited to linear transformations of the data, meaning non-linear relationships between features cannot be captured. Another more sophisticated and non-linear technique that can be used is called Uniform Manifold Approximation and Projection (UMAP). UMAP has recently been demonstrated as an effective pre-clustering dimensionality reduction technique (Allaoui et al., 2020; Asyaky & Mandala, 2021).

A recent review highlighted many different algorithms for semi-supervised clustering (Qin et al., 2019). Many of these are adaptations of the k-means clustering algorithm (Bair, 2013). Selecting an algorithm to use, however, will be limited based on the statistics/machine learning platform used and the statistics and coding skills of the analyst. Based on the authors observations, there very few off-the-shelf implementations of semi-supervised clustering available. The authors were only able to identify one readily available for the python programming language, for example (Babaki, 2017). Depending on the skill of the analyst, coding an algorithm from scratch may be an option.

## Model fitting and hyperparameter optimization

Cluster validation is the act of verifying cluster goodness-of-fit to the “true” clusters in the data based on internal or external criteria (Rendón et al., 2011). Typically, cluster validity metrics are used to select the algorithm and tune algorithm hyperparameters, most important being the number of clusters.

Internal cluster validation seeks to evaluate cluster results based on preconceived notions of what makes a “good” cluster, typically measuring qualities such as cluster compactness, cluster separation, and connectedness between points (Rendón et al., 2011). There are a variety of internal cluster validation metrics, including silhouette index, the Calinski–Harabasz index, and the Dunn index. These metrics can be limited because they make assumptions about the shape of clusters, and therefore may be biased towards certain algorithms.

External cluster validation measures the difference between a newly generated cluster structure and a ground truth cluster structure (Wu et al., 2009). This can consist of manual evaluation of cluster contents by analysts, or with the use of quantitative metrics. Example metrics include the rand index, mutual information-based scores,  $F$  measures, and cluster purity. Selecting a clustering algorithm and parameters using external validity ensures that cluster solutions are optimized based on a ground truth preferred by the analyst, however, this requires labeled data.

Ideally, cluster results should be evaluated using a mix of internal and external validation, whether this is quantitative or by analyst review (Gajawada & Toshniwal, 2012). This can prevent overreliance on preconceptions about the organization of the data while taking advantage of prior subject matter expertise.

To generate cluster results to evaluate, hyperparameters should be systematically varied. Two of the most common approaches for hyperparameter optimization are grid-search and random-search. For grid-search, hyperparameters values are generated equally-spaced within their respective bounds, and a model is fit for each combination of these generated values. For random-search, each set of hyperparameters is generated within the given bounds using some random process. While grid-search approaches guarantee that all areas of the defined parameter-space are searched, random-search approaches are typically more efficient (Bergstra & Bengio, 2012).

## Clustering evaluation and selection

Quantitative metrics alone are simple but may not result in the best solution. This work suggests a process for manual evaluation of a subset of cluster results to select a best solution. To manually evaluate document cluster results, the resulting clusters must be coherently summarized.

Automatic summarization and interpretation of document clusters is a significant and ongoing challenge due to the complex and high-dimensional nature of text-based data (El-Kassas et al., 2021). In this method, we suggest the following process that can be applied relatively easily using existing NLP-capable software. First, “top words” for each candidate cluster result are generated using TF-IDF. TF-IDF is an algorithm that represents each document in a corpus as a vector with each entry corresponding to each unique word or phrase in the corpus. The values in the vector correspond to the frequency of each word in the document weighted by the inverse frequency of the word throughout the entire corpus. Thus, words that appear frequently in a document are considered important to that document, but words found commonly throughout the corpus are weighted as less important. Equations 1–3 shows the scikit-learn default implementation of TF-IDF (Pedregosa et al., 2011).

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, d, D) \quad (1)$$

$$TF(t, d) = f_{t,d} \quad (2)$$

$$IDF(t, d, D) = \log \frac{1 + |D|}{1 + |\{d \in D : t \in d\}|} + 1 \quad (3)$$

where  $f_{t,d}$  is the frequency of term  $t$  in document  $d$ .  $D$  is the set of all documents. Using TF-IDF, top words can be generated for clusters by creating TF-IDF vectors for each document in each cluster, and then computing mean or median values for each word. The highest value words associated with each cluster are considered important words for that cluster topic.

A rating system was devised to externally evaluate and select a final cluster result. The rating system criteria were inspired by the rating system used by Zhang et al. (2018). The criteria are term coherence, clustering distinctiveness, and relevancy. The guide for rating these criteria and corresponding values are as follows:

- Relevance—Do the cluster top words convey a topic related to the goals of the analysis?



- 0. No
  - 1. Yes
- Coherence—Is a meaningful and unified topic conveyed by the cluster top words?
    - 0. Incoherent—Words appear random with little relation to each other, are too general to discern a meaningful topic, or convey multiple, unrelated topics.
    - 1. Average—Some words are related and meaningful in relation to the identified topic.
    - 2. Good—Most words are related and meaningful in relation to the identified topic.
  - Distinctiveness—Is the conveyed cluster topic unique?
    - 0. Indistinct—Cluster topic is redundant with one or more topics.
    - 1. Partially Distinct—Cluster topic is partially related to one or more topics.
    - 2. Distinct—Cluster topic is completely unique.

To utilize these criteria, several top words should be generated for each cluster using the TF-IDF approach above. For each cluster, the raters should generate a topic name or description conveyed by the words. If a name cannot be conceived, or multiple topics are conveyed for the same cluster, then the cluster can be labeled incoherent. Then, the rater should evaluate if the cluster is relevant to the goals of the analysis. Next, the rater should evaluate the coherence of the top words. Last, the rater should examine all cluster topics and rate them based on their distinctiveness.

Prior to rating the candidate cluster results, raters should evaluate a cluster result that is not a candidate for selection. The results of these ratings should then be compared using percent agreement or some other metric for inter-rater reliability (Gisev et al., 2013). If the results demonstrate agreement, then rating of the candidate solutions can commence. If they do not demonstrate agreement, then the raters should discuss the results of the rating and try to identify sources of disagreement and edge-cases such that raters can be calibrated to one another. A second calibration rating should then be performed.

The goal of evaluating clusters should be to select a cluster solution that maximizes coherence and distinctness for as much of the dataset as possible. A score for each metric that weights ratings based on the size of each cluster they are applied to can be calculated as follows:

$$\text{Criteria Score} = \frac{\sum_{i=1}^k n_i \times r_i}{N \times \max(\{r : r \in R\})} \tag{4}$$

where  $k$  is the number of clusters,  $r_i$  is the rating associated with cluster  $i$ ,  $n_i$  is the number of documents associated with cluster  $i$ ,  $N$  is the total number of documents, and  $R$  is the set of possible ratings for that criterion. This should be applied to each criterion individually.

All three criteria are important and should be used simultaneously when selecting a final clustering result. While this may not always be the preference, it is the opinion of the authors that the order of priority should be domain relevance, coherence, and distinctiveness. If a set of clusters are highly coherent and distinct, but do not answer the questions of the analyst, then they are not useful. Further, while indistinct (i.e., redundant) clusters are generally not ideal, they can still be useful, and even expected. Documents that humans understand as conceptually similar may still be separated in the

embedding-space if the applied embedding does not capture the similarity between the words within each document. It may therefore be inevitable that some clusters are indistinct based on the limitations of existing language models. These clusters can easily be merged post-evaluation and selection. On the other hand, an incoherent cluster cannot be salvaged or made useful.

## Demonstration of method

### Description of dataset

The U.S. Army Medical Research and Development Command's (USAMRDC) Telemedicine and Advanced Technology Research Center (TATRC) manages the Advanced Medical Technology Initiative (AMTI), an intramural program that supports innovative ideas from military and government civilians assigned to military treatment facilities and provides them with small funding investments to explore military healthcare performance improvements and technology demonstrations. AMTI annually solicits research proposals from these parties to evaluate and select candidate funding recipients. Proposals are evaluated on the quality of the concept, the relevance to military healthcare, the rigor and validity of the proposed methods, and the potential return on investment.

The data used for this demonstration are AMTI preproposals from 2010 to 2022 ( $n = 825$ ). Each pre-proposal includes the following mandatory free-text sections: Short Description, Deliverables, Alignment with Identified Gap, Problem to be Addressed, Military Relevance, Potential Impact on Military Health System (MHS), Transition Plan, Technology to be Demonstrated, Significance/Impact, Metrics to be Used, Personnel, and Partner Institutions. AMTI was specifically interested in categorizing proposals by the problems they sought to address, referred to as “problem-sets”, to track trending research topics. The focus of this analysis was therefore the “Problem to be Solved” section of the AMTI proposal template. The median, 5th, and 95th percentile word count for these sections were: 463 (93.6, 1112.6) words. By demonstrating this method on a real-world data set, we can show how our generalized framework can be molded to a specific application while providing guidance on challenges and pitfalls that we encountered that may be encountered by those seeking to replicate. While this data set does not represent every conceivable data set this method could be applied to, these results should generalize well to proposal data sets of similar length and composition.

### Manual annotation of demonstration texts

Proposals from years 2021–2022 were sampled from the data set for manual annotation. This sample included 123 proposals, or ~15% of the data. As mentioned in Sect. “[Methodology](#)”, ideally documents should be randomly sampled throughout the data set to limit selection bias. In this case, the authors were required to manually label documents for the stated years for a competing project. Annotation of the documents was performed by two analysts corresponding to the paper authors. The process described in Sect. “[Manual document annotation](#)” was followed.

## Preparation of text

Preparation of text for clustering includes pre-processing, transformation of text to fixed-length vectors, and dimensionality reduction. Pre-processing steps depended on the embedding technique used, as discussed in Chapter 2.2.1. Prior to translating text to vectors, the mandatory proposal sub-section “Problem to be Solved” was isolated. Three different text vectorization techniques were applied: TF-IDF, GloVe, and BERT.

### TF-IDF

A description of the TF-IDF algorithm was provided in Sect. “[Clustering evaluation and selection](#)”. The Scikit-learn (Pedregosa et al., 2011) implementation of TF-IDF was used to create document vectors for each proposal. The ngram range was set to 2. The algorithm was also set to filter terms that occurred in more than 50% of the documents. This is a common practice prior to clustering or topic modeling to eliminate words that provide little discriminatory power and to reduce noise (Pourrajabi et al., 2014).

### GloVe

GloVe uses a log-bilinear regression model to generate weight vectors for words based on the probability of word-word co-occurrence in massive corpuses of text. The resulting vectors, or word embeddings, demonstrate contextual information with relation to one another that allow algebraic operations to be performed that preserve linguistic meaning (Pennington et al., 2014). There are several pretrained GloVe embeddings available at: <https://nlp.stanford.edu/projects/glove/>. In this work, the pretrained “Common Crawl” embedding is used. This embedding can represent 840B unique word tokens and results in vectors of 300 dimensions.

### BERT

BERT is one of the most recent and powerful efforts to create pre-trained language models to aid natural language processing tasks (Devlin et al., 2019). One of BERT’s biggest strengths compared to other models is the ability to capture contextual information bidirectionally for a word within a sentence (Cohan et al., 2019). The specific implementation of BERT used in this work is called Sentence-BERT (SBERT), a modification of BERT specifically tailored to produce sentence-level embeddings (Reimers & Gurevych, 2019). Note that a “sentence” in this context refers to a sequence of words and may include multiple “linguistic” sentences, punctuation included. The python framework “sentence-transformers” was used to implement SBERT and the pretrained model used was “all-distilroberta-v1”. This pretrained model was selected because it ranked high in performance in the SBERT documentation. The max sequence length for this model is 512 and the resulting number of embedding dimensions is 768. To accommodate the max sequence length, the first and last 256 words were concatenated together for each document.

## Model fitting and hyperparameter optimization

### Algorithms

In this method, we use the COP- $k$ -means algorithm (Wagstaff et al., 2001), an adaptation of the popular  $k$ -means algorithm that takes partially labelled data as input and tries to satisfy those labels while fitting clusters. COP- $k$ -means requires ML and CL constraints be specified prior to fitting to augment the fitting process. In this work, we elected to only utilize the ML constraints, allowing for the possibility that manually annotated categories belong in the same cluster given the global context. This may be an advisable approach too if computational resources are limited. We observed that algorithm runtime was associated with the number of specified constraints. To allow for comparison with a baseline approach, we also applied the method using the traditional  $k$ -means clustering algorithm. Because traditional  $k$ -means is completely unsupervised, step 1 (Fig. 1) of the method was disregarded. All other steps were performed as stated.

Prior to clustering, the data was dimensionally reduced with the UMAP algorithm using the python module “umap-learn” (McInnes et al., 2020). This implementation of UMAP includes several parameters that can potentially influence cluster output, referred to as nearest neighbors (NN), minimum distance (MD), and the number of components (NC) (McInnes et al., 2020). Details for these parameters are available in the umap-learn documentation (<https://umap-learn.readthedocs.io/en/latest/index.html>).

### Hyperparameter search

A hybrid random-search-grid-search approach to parameter selection was used for UMAP and COP- $k$ -means hyperparameters. 100 sets of UMAP parameters (NN, MD, NC) were randomly generated, and for each the number of clusters were varied sequentially from 20 to 100, resulting in 8100 different parameter combinations. Ranges for UMAP parameters were guided by umap-learn documentation and experimentation by the authors. A summary of the parameters used are shown in Table 1. For each combination of parameters,  $5 \times 5$  k-folds cross validation was used to examine unbiased performance of cluster external validation metrics. K-folds cross-validation is a technique where the data is split into  $k$  “folds”, the model is trained on  $k-1$  folds, and then tested on the remaining fold. This is repeated until each fold has been left out once. Cross-validation has been demonstrated as potentially viable for selecting semi-supervised cluster results (Pourrajabi et al., 2014).

**Table 1** Summary of model hyperparameters and ranges used for hybrid random-search-grid-search approach

Hyperparameter	Values	Type	Generation
Nearest Neighbors (NN)	[5,100]	Decimal	Random
Minimum Distance (MD)	[0,0.2]	Decimal	Random
Number of components (NC)	[2,30]	Integer	Random
Number of clusters	[20,100]	Integer	Grid (increment by 1)

## Cluster validation metrics

This work examines the use of two commonly used metrics for external validation: adjusted rand index (ARI) (Rand, 1971) and adjusted mutual information (AMI) (Gates & Ahn, 2017). Both metrics quantify the agreement between expected (ground truth) cluster labels and new cluster labels. Both metrics are “adjusted” for chance such that a value of 1 indicates perfect dependence between newly generated and ground truth labels, and 0 indicates complete independence between newly generated and ground truth labels (generated labels appear random). There are many metrics that can be used for internal clusters evaluation, however, in this work we use the popular silhouette index (SIL) (Starczewski & Krzyżak, 2015). Silhouette index measures cluster compactness, a measure of intra-cluster variance, along with cluster separation, or the distance between clusters (Brock et al., 2008). A higher value indicates dense and well-separated clusters, with a maximum value of 1.

## Hyperparameter search process and evaluation

Initial model fitting attempts revealed computation to be very time consuming (4.6–24.1 s/cluster run). With the 25 repeats per fitting ( $5 \times 5$  cross-validation)  $\times$  8100 hyperparameters  $\times$  3 embeddings, cross-validation for all hyperparameter values was consider infeasible. Instead, the following procedure was used.

First, the data was clustered for all 8100 different combinations of parameters for all three embedding techniques. AMI, ARI, and SIL were recorded for each clustering. It was observed that AMI and ARI were highly correlated ( $R^2$ : 0.986, SBERT; 0.969, GloVe; 0.968, TF-IDF) and therefore mostly redundant. The researchers opted to use AMI for remaining analyses. Second, for each embedding, the Pareto optimal cluster results were isolated. Pareto optimal, or Pareto efficient, refers to a solution that cannot be improved in one objective without worsening another. The set of solutions that are Pareto efficient is referred to as a Pareto front (M. Li et al., 2022). Pareto optimality was determined using maximum AMI and SIL, and minimum number of clusters. Third, each Pareto optimal combination of parameters are used to re-cluster the data, this time using  $5 \times$  fivefold cross-validation. AMI on each held-out fold was recorded, referred to as Testing AMI. AMI is also recorded for the retained data folds, referred to as Training AMI. Last, we select several candidate solutions amongst the Pareto efficient cluster results to move on to manual evaluation given several strategies. These strategies were: (1) highest AMI; (2) highest SIL; (3) highest test AMI (from cross-validation).

## Cluster evaluation and selection

Cluster evaluation and selection followed the process in Sect. “[Clustering evaluation and selection](#)”. There were 9 candidate solutions corresponding to each combination of text embedding and selection strategy for COP- $k$ -means, and 6 candidate solutions from standard  $k$ -means clustering (cross-validation not possible, therefore no testing AMI). Two experienced administrators and reviewers of grant proposals in military medicine served as raters. Raters were given thorough instruction for each rating criteria. Top words for each cluster were generated using the TF-IDF procedure described. Prior to rating candidate cluster results, the raters first independently reviewed and rated non-candidate cluster results until satisfactory agreement had been reached to attain calibration. Agreement was

measured using Cohen’s weighted kappa statistic. A statistic of 0.61 was determined to be the cut-off value for completion for each criterion (coherence, relevance, distinctiveness). This was based off the traditional guidance stating that  $\kappa > 0.61$  indicates a “substantial” level of agreement, or “moderate” agreement by other authors (McHugh, 2012). When satisfactory agreement had been reached, a single rater rated the remaining candidate solutions.

## Results

### Manual annotation

Both reviewers independently reviewed the 123 documents and generated a list of problem-set categories. The reviewers then convened, compared lists, and compiled a final list of 36 categories. The reviewers then independently applied the codes. Comparison of codes revealed a 64% rate of agreement. The reviewers discussed results, reconciled disagreements, and merged several categories. A final rule was made on disagreements, resulting in a final list of 23 categories (see Supplementary Material S1).

### Quantitative cluster performance

#### Cluster performance summary

Table 2 contains the mean and range for performance metrics.

SBERT produced better values on average for AMI and SIL. Further, COP- $k$ -means produced results with higher AMI but lower SIL on average compared to standard  $k$ -means. For both metrics, there was a notable relationship between the # of clusters and the resulting metrics. This relationship is visualized in Figs. 2 and 3.

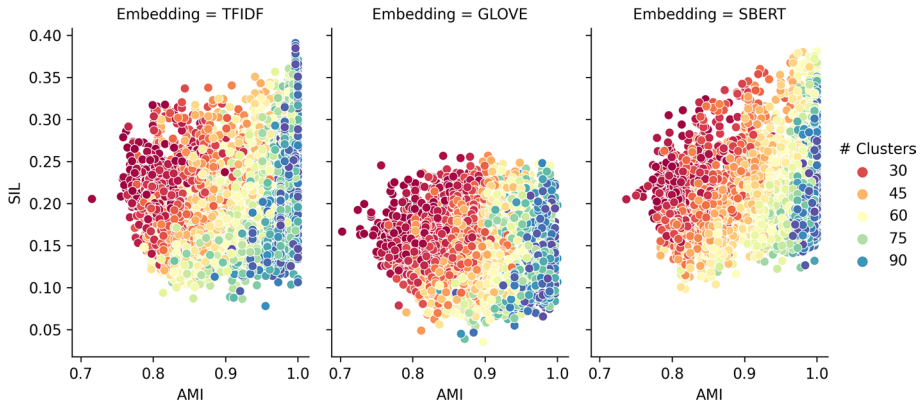
It can be clearly seen that there is an increasing trend between AMI and the number of clusters for the COP- $k$ -means algorithm. Conversely, this trend reversed when using the traditional  $k$ -means algorithm.

To examine the relationship between hyperparameters and performance metrics, ordinary least squares regression was applied. Only results for COP- $k$ -means were examined. Two models were fit with AMI and SIL as dependent variables and hyperparameters and embedding type as independent variables. Table 3 contains the results of

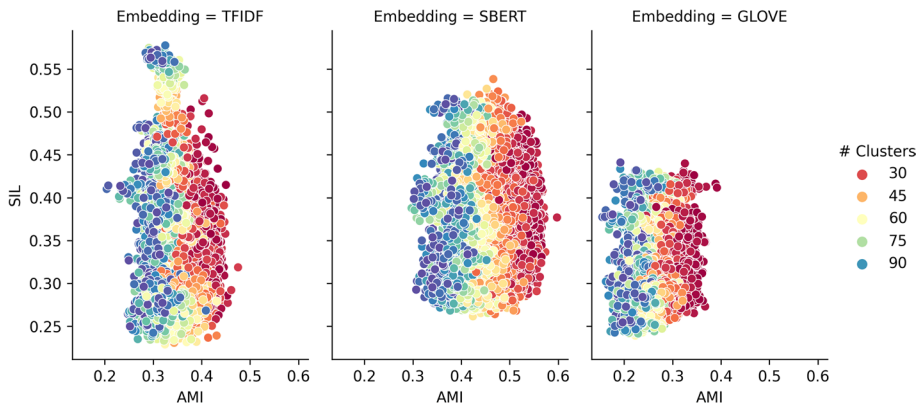
**Table 2** Summary of clustering fitting performance (mean and range) for each embedding and performance metric

		TF-IDF	GLOVE	SBERT
cop-km	Mean AMI (range)	0.900 (0.716,1.0)	0.910 (0.702,1.0)	0.915 (0.737,1.0)
	Mean SIL (range)	0.187 (0.078,0.391)	0.133 (0.036,0.256)	0.200 (0.098,0.381)
km	Mean AMI (range)	0.349 (0.204,0.475)	0.256 (0.155,0.392)	0.441 (0.297,0.598)
	Mean SIL (range)	0.332 (0.229,0.575)	0.304 (0.239,0.441)	0.354 (0.262,0.538)

*km*  $k$ -means; *AMI* adjusted mutual information; *SIL* Silhouette index; *GLOVE* global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency



**Fig. 2** Performance metrics plotted against the number of clusters fit for COP-k-means clustering results generated from hyperparameter optimization. *GLOVE* Global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency



**Fig. 3** Performance metrics plotted against the number of clusters fit for k-means clustering results generated from hyperparameter optimization. *GLOVE* global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* Term frequency-inverse document frequency

the regression analysis for AMI. Independent variables were min–max scaled to facilitate interpretation. Standard assumptions were checked (linearity of residuals, normality of residuals, homoscedasticity of residuals, multicollinearity) visually, and no large violations were detected.

Within the bounds of the independent variables, # of clusters was the most influential variable on average. Table 4 contains the results of the regression analysis for SIL. Visualization of the data prior to modeling indicated that there may be a quadratic relationship between both NN and the # of clusters with SIL. The model was fit both with and without squared terms for these variables, and both  $R^2$  and AIC performed better for the squared-term model. Standard assumptions were checked visually, and no large violations were detected.

**Table 3** Regression analysis for AMI given embedding used and model hyperparameters

Independent variable	Coefficient	Std. Error	<i>P</i> -value	$R^2$	F-stat	<i>n</i>
<i>Embedding (ref. = GLOVE)</i>	–	–	–	0.764	13,110	24,300
SBERT	0.003	< .001	< .001			
TF-IDF	– 0.012	< .001	< .001			
Nearest neighbors	– 0.029	0.001	< .001			
Minimum distance	– 0.007	0.001	< .001			
# of components	0.003	0.001	< .001			
# of clusters	0.188	0.001	< .001			
Intercept	0.834	0.001	< .001			

Independent variables were min–max scaled

AMI adjusted mutual information; *GLOVE* Global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency

**Table 4** Regression analysis for SIL given embedding used and model hyperparameters

Independent variable	Coefficient	Std. Error	<i>P</i> -value	$R^2$	<i>F</i> -stat	<i>n</i>
<i>Embedding (ref. = GLOVE)</i>	–	–	–	0.787	11,220	24,300
SBERT	0.064	< .001	< .001			
TF-IDF	0.050	< .001	< .001			
Nearest Neighbors	– 0.279	0.002	< .001			
Nearest Neighbors <sup>2</sup>	0.189	0.002	< .001			
Minimum Distance	– 0.035	0.001	< .001			
# of Components	– 0.005	0.001	< .001			
# of Clusters	– 0.081	0.002	< .001			
# of Clusters <sup>2</sup>	0.066	0.002	< .001			
Intercept	0.250	0.001	< .001			

Independent variables were min–max scaled

SIL Silhouette index; *GLOVE* Global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency

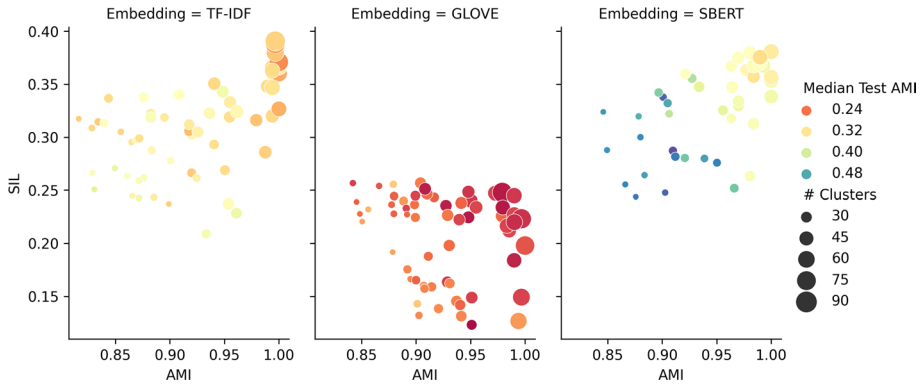
## Pareto analysis and cross-validation

Figure 4 shows the Pareto front for each embedding (COP-*k*-means only), color coded by the median test AMI value for each Pareto optimal result.

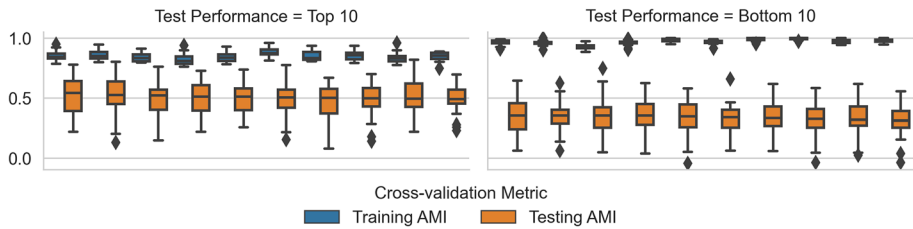
On average, test AMI was highest for SBERT. In general, a lower number of clusters corresponded to a higher test score.

Figure 5 shows the distribution of testing AMI for the top and bottom 10 performing combination of hyperparameters on the Pareto front when using SBERT. Top and bottom 10 performers were determined using the median test score. This plot shows that lower testing score corresponded with a relatively higher training score (which corresponded with a larger number of clusters). This trend was observed for the other embeddings as well.





**Fig. 4** Pareto optimal cluster results with the # of clusters indicated by point size and cluster cross-validation results indicated by color. *AMI* adjusted mutual information; *GLOVE* global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency. (Color figure online)



**Fig. 5** Cross-validation training and testing AMI for the top and bottom 10 performers with SBERT embedding. *AMI* adjusted mutual information

## Manual ratings

### Rater reliability

Raters 1 and 2 rated a non-candidate cluster result containing 61 clusters. The % agreement ( $P_a$ ), Cohen’s weighted kappa, and 95th confidence interval was calculated for each criterion. Coherence:  $P_a=65.6\%$ ,  $\kappa=0.31$  (0.05, 0.57), Relevance:  $P_a=77.5$ ,  $\kappa=0.13$  (– 0.27, 0.53); Distinctiveness:  $P_a=52.5$ ,  $\kappa=0.28$  (0.09, 0.47). This was determined to be unsatisfactory reliability, and the raters decided to repeat the process with another cluster result containing 28 clusters. Prior to rating, the raters identified and discussed discrepancies. Further, to assist in interpretability, each top word was colored coded on a red-yellow-green gradient based on the proportion of documents in the cluster that contained that word at least once. The results were—Coherence:  $P_a=85.7\%$ ,  $\kappa=0.75$  (0.51, 0.99), Relevance:  $P_a=96.4\%$ ,  $\kappa=0.65$  (0.02, 1.32); Distinctiveness:  $P_a=82.1\%$ ,  $\kappa=0.77$  (0.57, 0.97). While  $\kappa$  was below threshold for relevance, % agreement was very high, so the rating process proceeded.

The candidate cluster results selected for manual rating are summarized in Table 5.

The ratings for the candidate cluster results are shown in Table 6.

**Table 5** Candidate cluster results by selection strategy (columns) and embedding (rows)

Model	Max AMI						Max SIL						Max testing AMI											
	AMI (train)		AMI (test)		SIL		NC		AMI (train)		AMI (test)		SIL		NC		AMI (train)		AMI (test)		SIL		NC	
	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC	AMI	NC
COP-KM	TF-IDF	1.0	61	0.276	0.327	0.327	61	0.996	0.302	0.391	0.391	94	0.831	0.389	0.389	0.251	20	0.831	0.389	0.389	0.251	20	0.251	20
	GLOVE	1.0	88	0.217	0.198	0.198	88	0.904	0.245	0.258	0.258	40	0.879	0.297	0.297	0.256	25	0.879	0.297	0.297	0.256	25	0.256	25
	SBERT	1.0	53	0.371	0.338	0.338	53	1.0	0.326	0.381	0.381	61	0.910	0.540	0.540	0.287	28	0.910	0.540	0.540	0.287	28	0.287	28
KM	TF-IDF	0.475	28	-	0.318	0.318	28	0.291	-	0.575	0.575	51	-	-	-	-	-	-	-	-	-	-	-	-
	GLOVE	0.392	21	-	0.411	0.411	21	0.326	-	0.441	0.441	22	-	-	-	-	-	-	-	-	-	-	-	-
	SBERT	0.598	25	-	0.377	0.377	25	0.466	-	0.538	0.538	40	-	-	-	-	-	-	-	-	-	-	-	-

SIL Silhouette index; AMI adjusted mutual information; KM k-means; GLOVE global vectors for word representation; SBERT sentence bidirectional encoder representations from transformers; TF-IDF term frequency-inverse document frequency

**Table 6** Ratings for candidate cluster results for each selection strategy (columns), rating criteria, and embedding (rows)

Model		COP-KM			KM	
		Max AMI	Max SIL	Max testing AMI	Max AMI	Max SIL
Coherence	<i>TF-IDF</i>	0.659	0.738	0.515	0.572	0.631
	<i>GLOVE</i>	0.626	0.638	0.526	0.398	0.427
	<i>SBERT</i>	0.732	0.789	0.656	0.515	0.580
Relevance	<i>TF-IDF</i>	0.872	0.914	0.744	0.695	0.835
	<i>GLOVE</i>	0.782	0.787	0.773	0.685	0.572
	<i>SBERT</i>	0.890	0.878	0.956	0.841	0.799
Distinctiveness	<i>TF-IDF</i>	0.294	0.265	0.561	0.300	0.184
	<i>GLOVE</i>	0.317	0.324	0.488	0.441	0.433
	<i>SBERT</i>	0.432	0.378	0.504	0.393	0.232

*SIL* Silhouette index; *AMI* adjusted mutual information; *KM* k-means; *GLOVE* global vectors for word representation; *SBERT* sentence bidirectional encoder representations from transformers; *TF-IDF* term frequency-inverse document frequency

Max SIL with SBERT yielded the highest coherence rating. K-means produced results that were generally less coherent but more distinct, similar to Max Testing AMI results for COP-k-means.

The final clustering results are too numerous to show here. Instead, Table 7 summarizes 8 clusters taken from SBERT Max AMI cluster solution with a high coherence rating. This includes the top 10 terms associated with each cluster, the name ascribed to the cluster, and the number of documents associated with the cluster. This solution was selected because it presented a good balance between high coherence and distinctiveness. The other cluster summaries and example ratings for this result are shown in the supplemental material (Supplementary Material S2).

## Discussion

### Manual annotations

Initial document annotation following generation of categories yielded moderate levels of agreement, however, disagreements facilitated meaningful discussions that helped refine categories. One particularly impactful source of disagreement was categories that overlapped too much to be applied consistently given the available information. To manage this, several categories were merged. For example, “Database Development”, “Data Management”, and “Data Distribution” were initially separate categories, but were merged into a single “Data Management and Distribution” category. A second source of disagreement was passages of text that could reasonably belong to more than one distinct category. For example, “Musculoskeletal Injury” and “Physical Performance and Movement” were often applied to the same text. These terms describe often overlapping but distinct concepts. In these cases, the annotators made a final rule based on what they considered the dominant theme. There is no hard rule for determining this, and annotators should leverage their domain expertise to make this judgement.

**Table 7** Example clusters for the SBERT MAX AMI cluster solution with high coherence ratings

Applied Name	Top 10 Words	Cluster size
Musculoskeletal injury	Runner, knee, running, injury, greater, foot, force, nrfs, avlr, stress fracture	28
Health data management	Information, health, record, data, ahlia, health information, system, hie, clinical, access	19
Mental and behavioral health	Stress, game, health, depression, symptom, behavioral, ptsd, mental, dog, psychological	31
Infectious disease	Covid, pathogen, covid 19, sars, vaccine, infection, disease, coronavirus, antibody, respiratory	18
Training and simulation	Training, simulation, skill, based, exercise, virtual, environment, casualty, medic, education	17
Pain management	Pain, lbp, back, back pain, disability, muscle, pelvic, chronic, outcome, cpps	16
Diabetes	Diabetes, health, mobile, data, type, a1c, medication, self-management, diabetic, blood	14
Cancer	Cancer, survivorship, plan, oncology, cancer patient, follow, repository, survivorship plan, melanoma, provider	5

*nrfs* non-rear foot strike; *avlr* average vertical loading rates; *ahlia* armed forces health longitudinal technology application; *hie* healthcare information exchange; *ptsd* post-traumatic stress disorder, *sars* severe acute respiratory syndrome, *lbp* lower-back pain, *cpps* chronic pelvic pain syndrome

## Analysis of cluster validity metrics

In general, COP-*k*-means produced results with lower internal validity (SIL), but much higher external validity (AMI) compared to *k*-means. That COP-*k*-means produced higher values for external validity metrics is unsurprising given that its objective function uses this external information in the fitting process. Whether internal or external validity is more valuable at this stage is difficult to say and is up to the individual analyst to determine. This methodology sought to provide guidance regarding this by evaluating several selection policies using a qualitative rating system, discussed later.

Two linear models were fit with AMI and SIL as dependent variables for performance data associated with COP-*k*-means. In both cases, the models accounted for a large portion of the variability of observed performance values, however, not all was accounted for (~24% for AMI, ~21% for SIL). This could partially be attributed to some undetected interaction between independent variables. This variability may also be due to the potential for the *k*-means algorithm to converge to local optima spurred by random initialization (Bair, 2013).

For AMI, SBERT produced the highest values on average. It can also be seen that the number of clusters was, by far, the most influential predictor of performance. Within the parameter space explored, “minimum distance” and the “number of components” had only marginal effects while the “nearest neighbors” parameter was estimated to have a maximum average influence of ~0.03 AMI, representing about 10% of the observed range of values.

For SIL, SBERT again provided the highest values on average. Again, “minimum distance” and “number of components” had relatively little influence on performance. It seems, therefore, that these two parameters may be de-prioritized in hyperparameter optimization. This is beneficial from a resource perspective, as the number of components had a significant impact on model convergence time. There was an interesting relationship between SIL and the number of clusters and nearest neighbors where, as they increased, SIL decreased, though with diminishing effect as values increased. The “nearest neighbors” parameter had the largest effect on SIL on average. Given this and the prior insights, it seems that this parameter and the number of clusters should be prioritized in hyperparameter selection.

Figures 2 and 3 visualizes the distribution of AMI and SIL observed during model training. These results demonstrate that with a thorough search, there was not a strong trade-off between internal and external cluster validation for both algorithms. Regardless, a Pareto analysis of the data allows one to navigate trade-offs that do exist amongst the best performing results when simultaneously considering both metrics and is a suggested practice.

## Pareto analysis and cross-validation

Isolating the Pareto optimal data points for each embedding resulted in many candidate cluster results. To obtain further data to inform selection of a result, cross-validation was performed on each Pareto optimal combination of hyperparameters. These results are visualized in Figs. 4 and 5. Figure 4 shows that amongst the Pareto front, high testing AMI data points tended to result from lower numbers of clusters, and generally coincided with lower training AMI. Figure 5 further demonstrates this, where data with the highest testing performance tended to coincide with lower training performance. Training performance tended to be more

representative of testing performance in these cases. This may indicate that optimizing purely based on AMI without hold-out may lead to overfitting, particularly as the number of clusters increases.

### Candidate cluster ratings

Initial cluster ratings yielded relatively low reliability, however, this was improved significantly on the second round of rating following discussion and improvement of the cluster results presentation. Relevance was particularly problematic during both rounds due to the unbalanced distribution of ratings. The large discrepancy between  $P_a$  and  $\kappa$  for relevance in the second round of ratings is due to the prevalence of ratings=1 (27 and 26, respectively) compared to ratings=0 (1 and 2, respectively) by both raters. This is known as prevalence and is measured with the prevalence index (Sim & Wright, 2005). In this case, despite the low  $\kappa$ , there was only one cluster that was disagreed on by the raters. This is because Cohen's kappa controls for chance agreement, and when the "true" prevalence of a class is very low, then chance agreement becomes very high, as was the case here.

While differences in ratings between embedding and strategy cannot be verified as statistically significant due to the low sample size, there were some general trends that emerged. SBERT, as expected, produced clusters with the highest coherence. Further, the highest coherence was observed for the Pareto optimal cluster result with the highest SIL. For relevance and distinctiveness, SBERT and TF-IDF generally contained results that were higher than GloVe. It is unclear why GloVe underperformed the other embeddings in terms of ratings. Referring to Sect. "Cluster performance summary", GloVe also underperformed on average in terms of SIL. Silhouette score rewards cluster compactness and cluster separation, so perhaps the poor distinctiveness ratings are due to relatively poorer cluster separation.

With respect to selection strategy, again speaking generally, Max SIL produced the highest coherence ratings, followed by Max AMI, followed by Max Testing (cross-validated) AMI. For distinctiveness, the opposite was true. Further, Max SIL resulted in cluster results with large numbers of clusters (avg. 65) relative to Max Testing AMI (avg. 24.3) and standard k-means clustering (31.2). Fewer clusters led to more distinct but less cohesive clusters. This seems to suggest that conceptually similar clusters exist separately in the document embedding space, and that not fitting enough clusters will blend coherent micro-clusters into larger, less coherent clusters. This separation of conceptually similar documents is likely due to the limitations of current language models that cannot yet capture the entire variability of language used to describe similar topics in all domains. This could also be a testament to the variability of writing styles present in a collection of proposal documents. Additional uniformity in proposal requirements that promotes more uniform writing may result in coherent document clusters that are easier to extract. In this case, it seems that using the maximum testing AMI to select results prevented overfitting from a quantitative perspective but underfits the data with respect to expert domain knowledge. Therefore, one may opt for higher-cluster solutions, with the expectation of merging clusters that are conceptually similar but separated in embedding space.

## Implications

### Practical implications

In this manuscript we described a methodology for partial automation of research document categorization in enough detail to support replication by practitioners. The method is first described in general, such that it can be easily replicated beyond the specific domain in the demonstration. Guidance and best-practices are also provided alongside each step. Many features of the method were tested and compared during the demonstration (semi-supervised vs. unsupervised cluster, document embedding technique, cluster result selection strategy) to provide practical guidance for replication and to contribute to the general body of knowledge regarding application of these techniques. Based on the results we observed, we suggest practitioners replicate this method using SBERT embeddings, and select a cluster solution by performing a Pareto analysis of the cluster metrics and taking the maximum SIL. Further, if time is available, practitioners should opt for higher-cluster solutions with the expectation of merging conceptually similar clusters post-analysis.

### Theoretical implications

This is the first research to describe a step-by-step framework that merges semi-supervised clustering, subject matter expertise input, and internal and external validity to achieve semi-automated research proposal categorization. This framework can serve as a point of reference for researchers wishing to improve this method or propose their own methods for similar objectives.

### Limitations and future work

One limitation of this work with respect to the overall objective is that it does not guarantee perfect categorization. Clustering minimizes manual human effort, but it potentially sacrifices the accuracy of categorization. This method sought to provide a rigorous step-by-step process to create document clusters that optimizes this trade-off, but there will no doubt be misclassifications, as with any other existing approach. Another significant limitation of this work was the lack of available annotators and raters. Additional annotators and raters could have provided additional methodological validity. Another limitation was the non-random sample of proposals for annotation, despite the recommendations in Sect. “[Manual document annotation](#)”. This was unavoidable, as discussed in Sect. “[Manual annotation of demonstration texts](#)”. A final limitation was that this method was only demonstrated on a single dataset. The generalizability of the findings in this paper needs to be verified on other data.

There is significant room for improvement to the methodology and additional validation of its constructs. Methods for improving the interpretability of text-based clusters are needed, particularly in cases where clusters are numerous and cluster summaries must be succinct but representative to facilitate efficient review. Average TF-IDF values to identify top-words worked reasonably well to represent the contents of proposals belonging to the same cluster, however, it was occasionally susceptible to over-weighting terms that occurred frequently relative to the entire corpus but only occurred in a minority subset of

cluster proposals. The median value may have been a better choice to ensure at least 50% of the cluster documents contained the highly weighted term. Color-coding terms based on their distribution across all proposals in the cluster helped identify outliers.

An interesting addition to this methodology would be an approach to compare the distribution of proposal content within a portfolio to the corresponding body of published manuscripts. Unfunded research proposals and published manuscripts represent two different stages of the research supply chain. Comparing distributions of content between these two could reveal discrepancies in demand for research as determined by funding agencies and publishing bodies (published manuscripts) and supply at the point of the researcher (proposals).

## Conclusion

In this work, a multi-stage, semi-supervised method to cluster and extract insights from legacy proposal documents was proposed and demonstrated. The output of this methodology are thematically similar proposal document clusters. A comparative analysis was performed, and several key insights were attained. First, it was demonstrated that cutting-edge text-embedding techniques can outperform legacy techniques, and they require similar or even less effort to apply. Further, several strategies for cluster result selection were demonstrated. It was observed that a mixed prioritization of internal and external cluster validity can lead to good results. Last, it was shown that semi-supervised clustering can produce qualitatively more coherent clusters with little trade-off in cluster distinctness compared to unsupervised clustering. Archives of administrative documents kept by various funding institutions could contain valuable insights to help optimize administrative operations. Researchers must continue to develop automated processes and tools to unlock the insights currently hidden away in these archives.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s11192-023-04689-3>.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

## References

- Afzali, M., & Kumar, S. (2019). Text Document Clustering Issues: and Challenges. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 1, 263–268 <https://doi.org/10.1109/COMITCon.2019.8862247>
- Allaoui, M., Kherfi, M. L., & Cheriet, A. (2020). Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study. In A. El Moataz, D. Mammass, A. Mansouri, & F. Nouboud (Eds.), *Image and Signal Processing* (pp. 317–325). Springer International Publishing.
- Almeida, F., & Xexéo, G. (2019). Word Embeddings: A Survey. <http://arxiv.org/abs/1901.09069>
- Arnarsson, I. O., Frost, O., Gustavsson, E., Jirstrand, M., & Malmqvist, J. (2021). Natural language processing methods for knowledge management-applying document clustering for fast search and grouping of engineering documents. *Concurrent Engineering*, 29(2), 142–152. <https://doi.org/10.1177/1063293X20982973>
- Asyaky, M. S., & Mandala, R. (2021). Improving the Performance of HDBSCAN on Short Text Clustering by Using Word Embedding and UMAP. 2021 8th International Conference on Advanced Informatics:



- Concepts, Theory and Applications (ICAICTA), 1–6. <https://doi.org/10.1109/ICAICTA53211.2021.9640285>
- Babaki, B. (2017). COP-Kmeans version 1.5. <https://doi.org/10.5281/zenodo.831850>
- Bair, E. (2013). Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews. Computational Statistics*, 5(5), 349–361. <https://doi.org/10.1002/wics.1270>
- Bajpai, S., Bajpai, R., & Chaturvedi, H. (2015). Evaluation of inter-rater agreement and inter-rater reliability for observational data: An overview of concepts and methods. *Journal of the Indian Academy of Applied Psychology*, 41, 20–27.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10), 281–305.
- Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2022). Legal case document similarity: You need both network and text. *Information Processing & Management*, 59(6), 103069. <https://doi.org/10.1016/j.ipm.2022.103069>
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit* (1st ed.). O'Reilly Media.
- Boyack, K. W., Smith, C., & Klavans, R. (2018). Toward predicting research proposal success. *Scientometrics*, 114(2), 449–461. <https://doi.org/10.1007/s11192-017-2609-2>
- Brock, G., Pihur, V., Datta, S., & Datta, S. (2008). cValid: An R package for cluster validation. *Journal of Statistical Software*. <https://doi.org/10.18637/jss.v025.i04>
- Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Currents in Pharmacy Teaching and Learning*, 10(6), 807–815. <https://doi.org/10.1016/j.cptl.2018.03.019>
- Choi, S., & Jun, S. (2014). Vacant technology forecasting using new Bayesian patent clustering. *Technology Analysis & Strategic Management*, 26(3), 241–251. <https://doi.org/10.1080/09537325.2013.850477>
- Cohan, A., Beltagy, I., King, D., Dalvi, B., & Weld, D. S. (2019). Pretrained Language Models for Sequential Sentence Classification. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3691–3697. <https://doi.org/10.18653/v1/D19-1383>
- Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management*, 57(2), 102034. <https://doi.org/10.1016/j.ipm.2019.04.002>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186 <https://doi.org/10.18653/v1/N19-1423>
- Dhanani, J., Mehta, R., & Rana, D. (2021). Legal document recommendation system: A cluster based pairwise similarity computation. *Journal of Intelligent & Fuzzy Systems*, 41(5), 5497–5509. <https://doi.org/10.3233/JIFS-189871>
- Ebadi, A., Tremblay, S., Goutte, C., & Schiffauerova, A. (2020). Application of machine learning techniques to assess the trends and alignment of the funded research output. *Journal of Informetrics*, 14(2), 101018. <https://doi.org/10.1016/j.joi.2020.101018>
- Edler, J., & Boon, W. P. (2018). ‘The next generation of innovation policy: Directionality and the role of demand-oriented instruments’—Introduction to the special section. *Science and Public Policy*, 45(4), 433–434. <https://doi.org/10.1093/scipol/scy026>
- El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165, 113679. <https://doi.org/10.1016/j.eswa.2020.113679>
- Freyman, C. A., Byrnes, J. J., & Alexander, J. (2016). Machine-learning-based classification of research grant award records. *Research Evaluation*, 25(4), 442–450. <https://doi.org/10.1093/reseval/rvw016>
- Gajawada, S., & Toshniwal, D. (2012). Hybrid Cluster Validation Techniques. In D. C. Wyld, J. Zizka, & D. Nagamalai (Eds.), *Advances in Computer Science, Engineering & Applications* (pp. 267–273). Springer. [https://doi.org/10.1007/978-3-642-30111-7\\_25](https://doi.org/10.1007/978-3-642-30111-7_25)
- Gates, A. J., & Ahn, Y.-Y. (2017). The impact of random models on clustering similarity. *The Journal of Machine Learning Research*, 18(1), 3049–3076.
- Ghasemi, Z., Khorshidi, H. A., & Aickelin, U. (2022). Multi-objective Semi-supervised clustering for finding predictive clusters. *Expert Systems with Applications*, 195, 116551. <https://doi.org/10.1016/j.eswa.2022.116551>
- Gisev, N., Bell, J. S., & Chen, T. F. (2013). Interrater agreement and interrater reliability: Key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, 9(3), 330–338. <https://doi.org/10.1016/j.sapharm.2012.04.004>

- Hu, Y., Milios, E. E., & Blustein, J. (2016). Document clustering with dual supervision through feature reweighting. *Computational Intelligence*, 32(3), 480–513. <https://doi.org/10.1111/coin.12064>
- Jiménez, P., Roldán, J. C., & Corchuelo, R. (2021). A clustering approach to extract data from HTML tables. *Information Processing & Management*, 58(6), 102683. <https://doi.org/10.1016/j.ipm.2021.102683>
- Kalpokaite, N., & Radivojevic, I. (2019). Demystifying qualitative data analysis for novice qualitative researchers. *The Qualitative Report*. <https://doi.org/10.46743/2160-3715/2019.4120>
- Kaya, K., Yılmaz, Y., Yaslan, Y., Ögüdücü, ŞG., & Çingir, F. (2022). Demand forecasting model using hotel clustering findings for hospitality industry. *Information Processing & Management*, 59(1), 102816. <https://doi.org/10.1016/j.ipm.2021.102816>
- Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., & Rudzicz, F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, 100, 100057. <https://doi.org/10.1016/j.yjbix.2019.100057>
- Kim, J., Yoon, J., Park, E., & Choi, S. (2020). Patent document clustering with deep embeddings. *Scientometrics*, 123(2), 563–577. <https://doi.org/10.1007/s11192-020-03396-7>
- Levine, C. S., Knisely, B., Johnson, D., & Vaughn-Cooke, M. (2022). A structured method to achieve cognitive depth for medical device use error topic modeling. *Human Factors in Healthcare*, 2, 100016. <https://doi.org/10.1016/j.hfh.2022.100016>
- Li, Y., Cai, J., & Wang, J. (2020). A Text document clustering method based on weighted BERT Model. *IEEE 4th Information Technology, Networking Electronic and Automation Control Conference (ITNEC)*, 1, 1426–1430. <https://doi.org/10.1109/ITNEC48623.2020.9085059>
- Li, M., Chen, T., & Yao, X. (2022). How to Evaluate solutions in Pareto-based Search-based software engineering? A critical review and methodological guidance. *IEEE Transactions on Software Engineering*, 48(5), 1771–1799. <https://doi.org/10.1109/TSE.2020.3036108>
- Liu, Q., Kusner, M. J., & Blunsom, P. (2020). A Survey on Contextual Embeddings. <http://arxiv.org/abs/2003.07278>
- Ma, J., Xu, W., Sun, Y., Turban, E., Wang, S., & Liu, O. (2012). An ontology-based text-mining method to cluster proposals for research project selection. *IEEE Transactions on Systems, Man, and Cybernetics - Part a: Systems and Humans*, 42(3), 784–790. <https://doi.org/10.1109/TSMCA.2011.2172205>
- McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3), 276–282.
- McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. <http://arxiv.org/abs/1802.03426>
- McNie, E. C. (2007). Reconciling the supply of scientific information with user demands: An analysis of the problem and review of the literature. *Environmental Science & Policy*, 10(1), 17–38. <https://doi.org/10.1016/j.envsci.2006.10.004>
- Mei, J.-P. (2019). Semisupervised fuzzy clustering with partition information of subsets. *IEEE Transactions on Fuzzy Systems*, 27(9), 1726–1737. <https://doi.org/10.1109/TFUZZ.2018.2889010>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*. (Vol. 26). Curran Associates Inc.
- Mishra, S. K., Saini, N., Saha, S., & Bhattacharyya, P. (2022). Scientific document summarization in multi-objective clustering framework. *Applied Intelligence*, 52(2), 1520–1543. <https://doi.org/10.1007/s10489-021-02376-5>
- Mittal, M., Goyal, L. M., Hemanth, D. J., & Sethi, J. K. (2019). Clustering approaches for high-dimensional databases: A review. *Wires Data Mining and Knowledge Discovery*, 9(3), e1300. <https://doi.org/10.1002/widm.1300>
- Mohammed, S. M., Jacksi, K., & Zeebaree, S. R. M. (2020). Glove word embedding and DBSCAN algorithms for Semantic document clustering. *International Conference on Advanced Science and Engineering (ICOASE)*. <https://doi.org/10.1109/ICOASE51841.2020.9436540>
- Molchanov, V., & Linsen, L. (2018). Overcoming the Curse of Dimensionality When Clustering Multivariate Volume Data. Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, (pp. 29–39) <https://doi.org/10.5220/0006541900290039>
- Mutasodirin, M. A., & Prasajo, R. E. (2021). Investigating text shortening strategy in BERT: Truncation vs summarization. *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2021, 1–5. <https://doi.org/10.1109/ICACSIS53237.2021.9631364>
- Nichols, L. G. (2014). A topic model approach to measuring interdisciplinarity at the National Science Foundation. *Scientometrics*, 100(3), 741–754. <https://doi.org/10.1007/s11192-014-1319-2>

- Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., & Dehak, N. (2019). Hierarchical transformers for long document classification. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019*, 838–844. <https://doi.org/10.1109/ASRU46091.2019.9003958>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(85), 2825–2830.
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (pp. 1532–1543) <https://doi.org/10.3115/v1/D14-1162>
- Penta, A., & Pal, A. (2021). What is this cluster about? Explaining textual clusters by extracting relevant keywords. *Knowledge-Based Systems, 229*, 107342. <https://doi.org/10.1016/j.knosys.2021.107342>
- Pourrajabi, M., Moulavi, D., Campello, R. J. G. B., Zimek, A., Sander, J., & Goebel, R. (2014). Model selection for semi-supervised clustering. *17th International Conference on Extending Database Technology (EDBT)*. <https://doi.org/10.5441/002/edbt.2014.31>
- Priya, D. S., & Karthikeyan, M. (2014). An efficient EM based ontology text-mining to cluster proposals for research project selection. *Research Journal of Applied Sciences, Engineering and Technology*. <https://doi.org/10.19026/rjaset.8.1118>
- Qin, Y., Ding, S., Wang, L., & Wang, Y. (2019). Research progress on semi-supervised clustering. *Cognitive Computation, 11*(5), 599–612. <https://doi.org/10.1007/s12559-019-09664-w>
- Rajput, K., & Kandoi, N. (2017). An ontology-based text-mining method to develop intelligent information system using cluster based approach. *International Conference on Inventive Systems and Control (ICISC), 2017*, 1–6. <https://doi.org/10.1109/ICISC.2017.8068581>
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*(336), 846–850. <https://doi.org/10.2307/2284239>
- Reddy, G. T., Reddy, M. P. K., Lakshmana, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access, 8*, 54776–54788. <https://doi.org/10.1109/ACCESS.2020.2980942>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, (pp.671–688). <http://arxiv.org/abs/1908.10084>
- Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of Computers and Communications, 5*(1), 27–34.
- Sadjadi, S. M., Mashayekhi, H., & Hassanpour, H. (2021). A two-level semi-supervised clustering technique for news articles. *International Journal of Engineering, 34*(12), 2648–2657. <https://doi.org/10.5829/ije.2021.34.12.C.10>
- Sandhiya, R., & Sundarambal, M. (2019). Clustering of biomedical documents using ontology-based TF-IGM enriched semantic smoothing model for telemedicine applications. *Cluster Computing, 22*(2), 3213–3230. <https://doi.org/10.1007/s10586-018-2023-4>
- Saravanan, R. A., & Babu, M. R. (2021). Information retrieval from multi-domain specific research proposal using hierarchical-based neural network clustering algorithm. *International Journal of Advanced Intelligence Paradigms, 19*(3–4), 422–437. <https://doi.org/10.1504/IJAIP.2021.116369>
- Sarewitz, D., & Pielle, R. A. (2007). The neglected heart of science policy: Reconciling supply of and demand for science. *Environmental Science & Policy, 10*(1), 5–16. <https://doi.org/10.1016/j.envsci.2006.10.001>
- Sim, J., & Wright, C. C. (2005). The Kappa statistic in reliability studies: Use, interpretation, and sample size requirements. *Physical Therapy, 85*(3), 257–268. <https://doi.org/10.1093/ptj/85.3.257>
- Starczewski, A., & Krzyżak, A. (2015). Performance evaluation of the Silhouette index. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *Artificial intelligence and soft computing* (pp. 49–58). Springer International Publishing.
- Subakti, A., Murfi, H., & Hariadi, N. (2022). The performance of BERT as data representation of text clustering. *Journal of Big Data, 9*(1), 15. <https://doi.org/10.1186/s40537-022-00564-9>
- Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification. In M. Sun, X. Huang, H. Ji, Z. Liu, & Y. Liu (Eds.), *Chinese Computational Linguistics* (pp. 194–206). Springer International Publishing. [https://doi.org/10.1007/978-3-030-32381-3\\_16](https://doi.org/10.1007/978-3-030-32381-3_16)
- Talley, E. M., Newman, D., Mimno, D., Herr, B. W., Wallach, H. M., Burns, G. A. P. C., Leenders, A. G. M., & McCallum, A. (2011). Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods, 8*(6), 443–444. <https://doi.org/10.1038/nmeth.1619>

- Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, 577–584.
- Wang, Y., Xu, W., & Jiang, H. (2015). Using text mining and clustering to group research proposals for research project selection. *48th Hawaii International Conference on System Sciences*. <https://doi.org/10.1109/HICSS.2015.153>
- Wu, J., Chen, J., Xiong, H., & Xie, M. (2009). External validation measures for K-means clustering: A data distribution perspective. *Expert Systems with Applications*, 36(3, Part 2), 6050–6061. <https://doi.org/10.1016/j.eswa.2008.06.093>
- Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H., & Zhang, G. (2018). Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics*, 12(4), 1099–1117. <https://doi.org/10.1016/j.joi.2018.09.004>
- Zhong, S. (2006). Semi-supervised model-based document clustering: A comparative study. *Machine Learning*, 65(1), 3–29. <https://doi.org/10.1007/s10994-006-6540-7>
- Zhou, Y., Lin, H., Liu, Y., & Ding, W. (2019). A novel method to identify emerging technologies using a semi-supervised topic clustering model: A case of 3D printing industry. *Scientometrics*, 120(1), 167–185. <https://doi.org/10.1007/s11192-019-03126-8>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Benjamin M. Knisely<sup>1</sup>  · Holly H. Pavlicsak<sup>1</sup>

✉ Benjamin M. Knisely  
benjamin.m.knisely2.ctr@health.mil

<sup>1</sup> Telemedicine and Advanced Technology Research Center, United States Army Medical Research and Development Command, Fort Detrick, MD 21702, USA