KeySI: An Interaction Framework for Tuning Text Embeddings Based on Human Feedback

Category: Research

Paper Type: Analytics Decisions

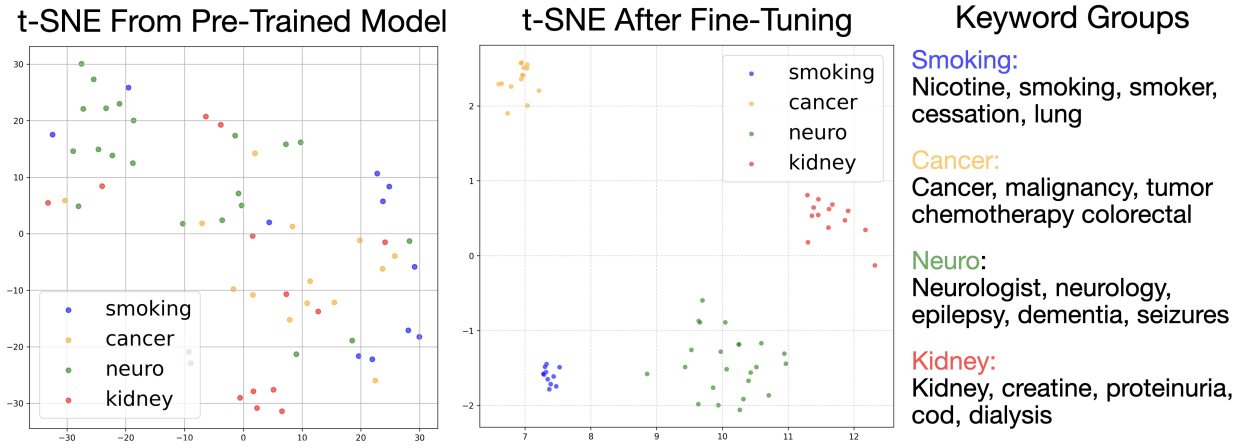

Fig. 1: A demonstration of KeySI on the COVID-19 data, which contains article abstracts for COVID-19 risk factors: smoking, cancer, neurological, and kidney. The initial embeddings (left) do not meaningfully capture the risk factors. Using KeySI, a user specifies keyword groups that define each risk factor (right) and tune the pre-trained model. The resulting embeddings (center) effectively capture and convey the risk factors to the pre-trained model.

Abstract—In large-scale text analysis tasks, pre-trained language models are often used to embed text corpora before performing analyses. However, pre-trained models may not effectively capture domain context and tuning them requires large amounts of labeled data and technical expertise to implement training pipelines. Recent approaches demonstrated the ability of visual interactions to capture learnable feedback and tune pre-trained models via direct organization of documents in a dimension reduction (DR) plot. However, these approaches require manual inspection of individual texts to locate relevant context. In this paper, we propose KeySI, an interactive deep learning framework that provides intuitive interactions in the textual feature space (via keyword extraction and organization) to specify domain information, translates the interaction into learnable model feedback in the document space, and tunes the model accordingly. By operating in the textual feature space, users no longer need to manually inspect or label documents or have the technical expertise to implement training pipelines. We present a prototype implementation of KeySI that employs DR to illustrate the state of the embedding space, automatically extracts and curates a set of keywords for the user, and enables interactive specification of keyword groups to convey domain context and tune the model. Finally, we validate KeySI through case studies and simulations, demonstrating its effectiveness in capturing and learning user feedback.

Index Terms—Text Data, Dimensionality Reduction, Semantic Interaction

1 Introduction

Recent years have seen a dramatic increase in the availability of pre-trained language models [40], across a variety of domains [14]. The emergence of these pre-trained models has resulted in a shift from experts creating small, task specific language models, to a "pre-training then fine-tuning" paradigm [40]. In this paradigm, rather than training from scratch with large amounts of data, users are able to fine-tune pre-trained models with smaller amounts of annotated data. This new paradigm significantly reduces the barrier of entry for employing language models - users do not need as much labeled data to effectively train for their task nor do they need as much technical expertise for implementing complex model pipelines. However, though the barrier of entry is lower, it may still be prohibitively high for non-expert users with little to no technical expertise for implementing training pipelines or curating labeled data for tuning [25].

To overcome these challenges, recent approaches have begun employing interactive visualization as an interface into DL models [3,4,21]. The visualization itself provides a window into the current state of the embedding space while accompanying interactions allow non-experts to intuitively convey their knowledge. These approaches then convert interactions into learnable feedback, thus tuning the model without requiring labeled data or deep technical expertise. For example, Bian and North's DeepSI [4] and Lin et al.'s ImageSI [21] allows users to interactively tune language and image models (respectively), relative to the domain information in the data. They rely on dimension reduction (DR) to visualize the current state of the embedding space, with respect to the data, and allow direct manipulation of the DR space to convey learnable feedback.

However, these approaches implicitly expect users to know what data items to manipulate to convey feedback. For images, directly plotting the image icons in the DR plot partially overcomes this limitation by cuing users to the contents of the data collection and specific instances to manipulate. For text, however, users must manually inspect individual documents to identify relevant texts for manipulation. This limits the utility of interactions directly in the document space, as manually searching through many documents for relevant information requires user overhead similar to manually labeling many documents.

To address this limitation, we propose KeySI, a method for inferring user feedback from intuitive interactions in the text *feature* space, rather

than the document space. KeySI provides a visual overview of the document embedding space using DR and an overview of the corpus feature space through a curated set of keywords. It allows users to interactively specify groups of keywords, from either the curated list or ad-hoc specification, that define concepts from their domain expertise. Then, using these keyword groups, KeySI identifies documents related to each group to use as "labeled" documents for tuning (the "labels" are implicit) and employs triplet margin loss [32] to tune the model accordingly.

To evaluate the effectiveness and robustness of the proposed method, we conducted systematic experiments on three datasets with distinct characteristics: COVID-19 risk factors, AG news, and BBC news. These datasets differ significantly in terms of scale, semantic structure, and the clarity of category boundaries. We evaluated KeySI from both qualitative and quantitative perspectives. In the qualitative analysis, we describe three usage scenarios that demonstrate the ability of KeySI to effectively capture user feedback. In the quantitative evaluation, we employ standard clustering metrics—Silhouette Score and Purity—to measure intra-cluster compactness and inter-cluster separability. We conduct a series of tests, such as parameter sensitivity analysis, to demonstrate the robustness of KeySI in capturing and incorporating user feedback to tune the embedding space, across the three datasets. In summary, the contributions of this research are as follows:

1. KeySI: an interaction method for capturing user feedback in the *text feature* space and translating this into learnable feedback in the document embedding space.
2. A prototype implementation of KeySI
3. Three usage scenarios demonstrating KeySI's ability to effectively capture feedback.
4. A quantitative evaluation of KeySIs sensitivity to various settings and parameters, demonstrating its ability to robustly incorporate user feedback.

2 Related Work

2.1 Text Sensemaking

Text sensemaking aims to help users efficiently extract insights and build a coherent understanding from large collections of textual documents. Previous works have primarily focused on supporting text sensemaking on interactions within the document space. For example, Kang et al. [17] studied the Jigsaw system [38], a visual analytics system for helping analysts who work with large collections of documents, highlighting the practical advantages and limitations of performing sensemaking tasks at the document and entity level. Meanwhile, Görg et al. [11] detailed the implementation of Jigsaw and highlighted how computational text analyses are smoothly integrated with interactive visualization to support document-level exploration.

In terms of user-driven exploration, Bradel, et al. [7] proposed a multi-model semantic interaction (MSI) framework, which enables users to directly manipulate documents within a visual layout to refine search queries and reorganize retrieved documents implicitly. Dowling et al. [8] introduced Cosmos, an interactive system emphasizing semantic interaction through spatial manipulations of document clusters. Liu et al. proposed TIARA [22] , which aims to integrate interactive visual summarization based on topic modeling and provides interaction tools for users to examine and interpret text collections from several perspectives.

While previous works require users to perform extra manual interactions in the document space, our approach focuses on the textual feature (keyword) space to allow users to directly convey domain knowledge through keyword selection and organization.

2.2 Interactive Model Training via User Feedback

Previous research has also investigated how to use user interaction to guide model improvement of text analysis results effectively. For example, early work by Sakai and Masuyama [31] demonstrated a multiple-document summarization system based on interactive user feedback at the keyword level. Their work highlights the strategy of users' feedback directly recalibrating the keyword weights through user selections. Another related work set the efforts to enable interactions at the semantic level. For instance, Kiefer et al.'s SemanticPush [18] suggest translating users' feedback into synthetic training examples that "push" models toward desired behaviors under text classification missions.

Some studies have instead focused on the impact of user interactions on model training through rule-based abstractions. HEIDL [35] enables the user to manipulate the rules from the model logic through direct interactions with rule predicates. RulesLearner from Yang et al. [41] introduced a structured rule-editing framework where users iteratively refine machine-learned semantic rules to construct trusted models. Mishra and Rzeszotarski [25] presented an interactive learning system designed for non-expert users to engage with transfer learning, and emphasized the feasibility of users' direct interaction in terms of iteratively neural network model's refinement.

Our work shares similarities with prior approaches in leveraging user feedback to refine our model behavior. More specifically, our method integrates human expertise during training by allowing users to define domain context via keyword clusters. Compared with previous works, including HEIDL and RulesLearner, our approach lets users interact with keyword groups directly in the textual feature space, which reduces the cognitive overload from rule formalisms or document-level manipulation. Moreover, most previous approaches rely on labeled data, while our approach adopts unsupervised embedding through triplet loss optimization that requires no labeled data as input.

2.3 Interactive Clustering

Several prior works have explored the possibility of implementing interactive feedback with the users to handle document clustering tasks from different perspectives. One common perspective is at the feature level. For example, Bekkerman et al. [2] introduced an interactive clustering approach that allows users to autonomously specify important features (e.g., sentiment-laden words) for desired groups and iteratively refine the clustering results through user interaction. Nourashrafeddin et al. [26] provides an algorithm that adopts interactive clustering concepts to iteratively cluster key terms rather than individual documents or single terms. At each iteration, the algorithm generates clusters of key terms for each document group and asks users to refine document clusters by reorganizing these key terms interactively. Hu et al. [15, 16] proposed a feature-level supervision framework based on user interaction. The system first automatically generates a feature list using an unsupervised feature selection method (such as the mean-TFIDF method) and displays the top features to the user for labeling. Then, the system will iteratively recompute document clustering based on user interaction.

Other works focus on visual analytics techniques for interactive clustering. Lee et al.'s iVisClustering [20] combined latent Dirichlet allocation (LDA) with interactive visualization techniques to provide users a direct way to iteratively optimize document clustering through interactions such as maintaining coherent clusters, removing clusters containing outliers, re-clustering, sub-clustering, and building hierarchical cluster structures. Moreover, Sherkat et al. [37] proposed a visualization analysis system that integrates coordinated visualization modules and document projection technology to achieve iterative clustering optimization through user-driven assignment of key terms and visualization operations. In later extension works, Sherkat et al. [36] enhanced this visualization analysis method by introducing coordinated visualization dedicated to studying document collections over time, which enables users to iteratively refine clusters by manipulating key terms based on domain knowledge. Rezaeipourfarsangi et al. [27] presented an interactive clustering system utilizing deep language models (e.g., SBERT and Universal Sentence Encoder) integrated with visual interfaces. Users can direct the clustering algorithm through the interface by interactively manipulating documents and clusters. Another related work proposed by Ruppert et al. [30] has implemented a visual interactive system for creating and validating document clustering workflows, which provides a visual interface for functions including feature selection, cluster analysis, and cluster comparison.

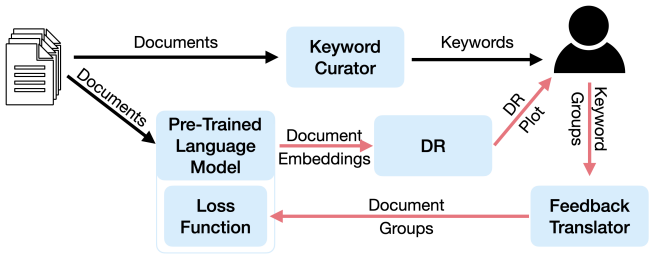

Fig. 2: An overview of the KeySI interaction framework. First, KeySI curates a set of representative keywords. Then it presents the keywords and a projection of the document embeddings to the user. The user then defines groups of keywords based on their domain expertise after which KeySI translates these keywords into learnable feedback and tunes the model. The red edges highlight the interactive learning loop.

Inspired by previous works, our work adopts a similar automated approach to generate keyword groups based on document features. However, our approach has two fundamental differences: (1) it does not require users to manually label or reorganize those generated features, which reduces the cognitive load on users for iterative manual adjustments and (2) the output is not only a clustering of documents be at tuned embedding space that enables further downstream analysis.

2.4 Semantic Interaction

Semantic interaction (SI) refers to the process that allows users to implicitly communicate analytical intent through visual manipulations and dynamically adapting visualization models [9]. Before our work, several studies have attempted to apply the concept of SI across diverse domains and interaction paradigms. For instance, DeepVA, proposed by Bian et al. [5], designed a visual analytics framework that bridges cognition and computation by replacing hand-crafted features with deep learning representations (e.g., BERT embedding). The study of DeepVA validates that high-level abstractions in deep features can better capture users' cognitive intent during semantic interactions, especially in high-abstraction tasks like document summarizations.

Extending DeepVA, Bian et al. [4] proposed DeepSI, which integrates deep learning techniques into the SI pipeline by fine-tuning BERT embeddings through spatial document rearrangements. NeuralSI [3] adopts a neural design concept to fill the gap that traditional dimensionality reduction methods cannot support several desired properties for visual analytics. The evaluation of the DeepSI and NeuralSI indicate that semantic interaction with high-level representations can accelerate the matching between the user's mental model and the computational embedding, especially in complex text analysis tasks.

In the image domain, Lin et al. introduced ImageSI [21], where users can directly manipulate 2D image projections to update the underlying model embeddings. ImageSI can dynamically optimize embeddings to capture task-specific semantics compared to traditional feature weighting methods.

In our work, we employed a similar approach to DeepSI of using deep learning representations to capture high-level semantics and enabling intuitive interaction to convey feedback. However, DeepSI relies on document-level spatial manipulations to convey feedback, thus requiring either prior knowledge of document contents (e.g., through labels) or manually searching through documents to locate relevant information. In contrast, our approach fills this gap by introducing keyword-level interaction in the textual feature space and eliminating the need for document-level spatial adjustments or labeled data.

3 KeySI

3.1 Framework Overview

Figure 2 gives an overview of the interaction framework. First, KeySI presents two views: a DR plot of the document corpus in the pre-trained embedding space and a curated set of keywords extracted from the corpus. Then, KeySI allows users to create groups of related keywords to define topics that will be given as feedback to identify groups of documents relevant to each topic. Finally, using these groups of documents as training data, the model employs Triplet Margin Loss [10, 33] to tune the embedding space with the provided feedback. After tuning, KeySI recomputes the DR to illustrate the updated embedding space. We describe each stage of the pipeline in more detail below.

3.2 Initial Projection and Keyword Curation

To provide an initial overview of the embedding space, KeySI projects the data using a DR method, shown in the left panel of Fig. 1. The prototype implementation uses t-SNE [39] to provide the initial projection, though the KeySI framework supports the use of any DR method.

Next, it curates and presents a set of keywords to describe the corpus. The baseline approach extracts the top keywords in each document and aggregates them across the corpus. However, as the data scales, this list quickly becomes long and unmanageable for users to navigate. To winnow down the space and guide users toward prominent themes we perform the following steps: (1) pre-process the documents to reduce the keyword search space, (2) extract and filter keywords, (3) embed and cluster the reduced set of keywords into topics, and (4) presenting the keywords through both projection and semantically clustered lists.

Pre-Processing KeySI first pre-processes each document to reduce the keyword search space and focus on terms with the most semantic meaning. It begins by removing unwanted characters and digits. Then, it uses the Natural Language Toolkit (NLTK) [6] to tokenize the document and perform part-of-speech tagging to identify nouns. We choose to extract nouns because they typically represent semantically rich sentence components, such as subjects and objects.

Extracting and Filtering Keywords Next, KeySI extracts the top keywords in each document. The prototype implementation uses KeyBERT [12] to extract the top n keywords, where n defaults to 3 but can be flexibly changed based on the corpus size. Again, while we chose KeyBERT because of its use of semantic similarities to identify keywords rather than word frequencies, the KeySI framework supports using any keyword extraction method. After extracting keywords, we alter keywords to preserve only their stem (the part of the word that carries the meaning). For example, the words 'smoke', 'smoking', and 'smoker' all share the same stem 'smoke'. To do this, we employ NLTK's implementation of the Porter Stemming algorithm [23]. Finally, by default three, KeySI filters the keywords based on an adjustable frequency threshold to help isolate significant keywords that likely represent recurrent themes. In other words, it maintains keywords across multiple documents rather than existing only in a single document. Stemming the keywords prior to filtering helps elevate groups of semantically similar words by combining their frequencies.

Embedding, Projection, & Clustering Lastly, though initial efforts aim to reduce the keyword space, the final result may still be a large set of keywords. To help guide users toward the topic, we embed the keywords into a high-dimensional vector representation, project these embeddings into 2D, and cluster the keywords into semantically related groups. The prototype presents these keywords to the user through either a color-encoded DR plot (see Fig. 4 (B)) or lists of keywords organized by semantic cluster.

Specifying Keyword Groups To specify keyword groups, the user selects keyword sets from the curated list to define sets of topics within the document corpus. The prototype provides some linked interactions to help connect the keywords back to the document projection, such as linked highlighting of documents containing a selected keyword or group of keywords. These linked interactions help the user identify how well their desired topics are represented by the current embedding space.

3.3 Translating Keywords into Learnable Feedback

The keyword groups defined by the user provide feedback in the text feature space. However, since the model operates in the document space, KeySI must translate the keyword groups into learnable feedback in the document space. To do this, KeySI automatically identifies

relevant documents and organizes them into disjoint groups based on the topic. This generates pseudo-classes of documents with implicit labels, suitable for tuning the model. KeySI uses a two-stage process for identifying relevant documents for each keyword group.

For example, to express the topic of smoking, a user selects a set of keywords such as "tobacco", "cigarette", and "nicotine". In the first stage, KeySI retrieves a broad set of candidate documents containing these keywords using a best matching algorithm. In the second stage, KeySI projects the candidates into a 2D space and performs semantic clustering and distance-based filtering to eliminate outliers and documents with semantically weak similarity to the keyword group. As shown in Fig. 3, only documents within a certain distance threshold to the cluster centroid are retained. The final result is a pseudo-class of semantically consistent documents suitable for model tuning.

3.3.1 Stage 1: Locating Candidate Documents

In the first stage, KeySI aims to identify pools of potentially similar documents. Given the flexible nature of the KeySI framework and the varying characteristics of our datasets, we adopt the Okapi BM25 algorithm [28] for initial document retrieval. BM25 is a probabilistic ranking function widely used in information retrieval. It incorporates document length normalization, which ensures stable and reliable relevance scoring even across corpora with inconsistent document lengths. This makes it particularly suitable for identifying a diverse pool of candidate documents, serving as a strong foundation for KeySI. To identify an initial pool of candidate documents relevant to each keyword group, we adopt the Okapi BM25 algorithm [28]. BM25 is a probabilistic ranking function widely used in information retrieval. It ranks documents based on term frequency (TF), inverse document frequency (IDF), and document length normalization. The relevance score between a document D and a keyword group query Q is computed as follows:

$$\text{score}(D,Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t,D) \cdot (k_1 + 1)}{f(t,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \quad (1)$$

Here, $f(t,D)$ is the frequency of term t in document D, $|D|$ is the length of document D, and avgdl is the average document length in the corpus. The inverse document frequency is computed as:

$$\text{IDF}(t) = \log\left(\frac{N - n_t + 0.5}{n_t + 0.5} + 1\right) \quad (2)$$

where N is the total number of documents, and n_t is the number of documents containing term t.

While BM25 efficiently retrieves documents containing relevant keywords, it does not consider the semantic context. Thus, some retrieved documents may be lexically similar but semantically unrelated. To address this problem, we introduce a second-stage semantic filtering based on contextual embeddings. However, this risk includes documents containing some query words but in different, unrelated contexts. As such, we perform an additional stage to filter the candidate pool down to the most semantically related subset.

3.3.2 Stage 2: Filtering Candidate Documents

To identify the latent semantic structures, we first project the document and keyword group embeddings into a 2D space. We explored the use of various DR methods, however methods like t-SNE [39] and UMAP [24] emphasize local structures, which result in overly compact or sparse clusters, impairing the reliability of spatial distance for filtering. Thus, We chose Multidimensional scaling(MDS) as it provides a more balanced and globally faithful projection, making distance-based filtering more meaningful. After projection, KeySI perform K-Means clustering in the 2D space to segment semantically coherent subgroups. It automatically determines the number of clusters by maximizing the Silhouette Score [29], ensuring both intra-cluster compactness and inter-cluster separation. Ideally, semantically relevant documents are expected to be grouped into a single dominant cluster. However, due to the lexical nature of BM25, it may retrieve outlier documents that include the keywords but convey different meanings—for instance, in the case of homonyms.

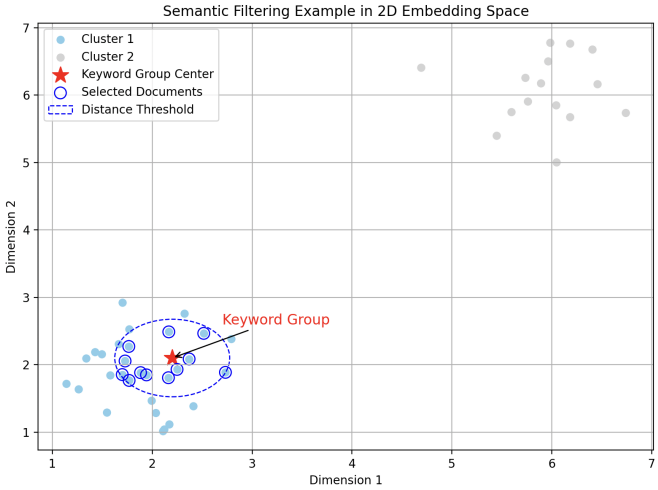

Fig. 3: An illustration of the final filtering step in the feedback translation process. KeySI filters candidate documents based on their distance to the keyword group center in the 2D space. Only documents within a specified percentile threshold are retained, ensuring that the final training set consists of samples with the strongest semantic alignment to the user's intent.

Next, KeySI computes the cosine distance between each cluster centroid and the 2D projection of the keyword group center (obtained by averaging the embeddings of the group's keywords). It then selects the cluster with the smallest cosine distance as the *primary semantic cluster* for that keyword group.

To further eliminate semantically marginal documents from the primary cluster, KeySI performs a final filtering based on the distance between each document in the primary cluster and the cluster centroid. It calculates the Euclidean distance between each document and the cluster centroid in the 2D space, and filters them based on a percentile threshold. Specifically, we define the distance threshold as the p-th percentile (e.g., 40%) of all distances in the cluster (see Fig. 3). Only documents that fall within this threshold are retained, resulting in a high-quality core subset that is more semantically aligned with the target keyword group.

While one could directly measure the distance from each candidate document to the keyword group center for filtering, this assumes that the candidate set forms a single coherent cluster. In reality, BM25-based retrieval often returns a mix of document groups with varying semantics—some containing keywords yet diverging from the intended topic.

Therefore, clustering serves as a crucial intermediate step for:

- identifying the sub-cluster most aligned with the keyword group's semantic center,
- filtering out documents that are lexically similar but semantically irrelevant,
- and reducing the impact of outliers on subsequent filtering.

In essence, clustering plays a dual role in semantic denoising and structural segmentation, enabling us to distill the candidate set into a more compact and consistent subset. This, in turn, enhances the quality and consistency of the pseudo-labeled documents used for downstream fine-tuning. By comparing the labels of document IDs before and after filtering, we observe that many documents with labels inconsistent with the intended keyword group were removed. This validates the effectiveness of the two-stage filtering process in eliminating semantically misaligned samples and preserving a focused, high-purity subset for model tuning.

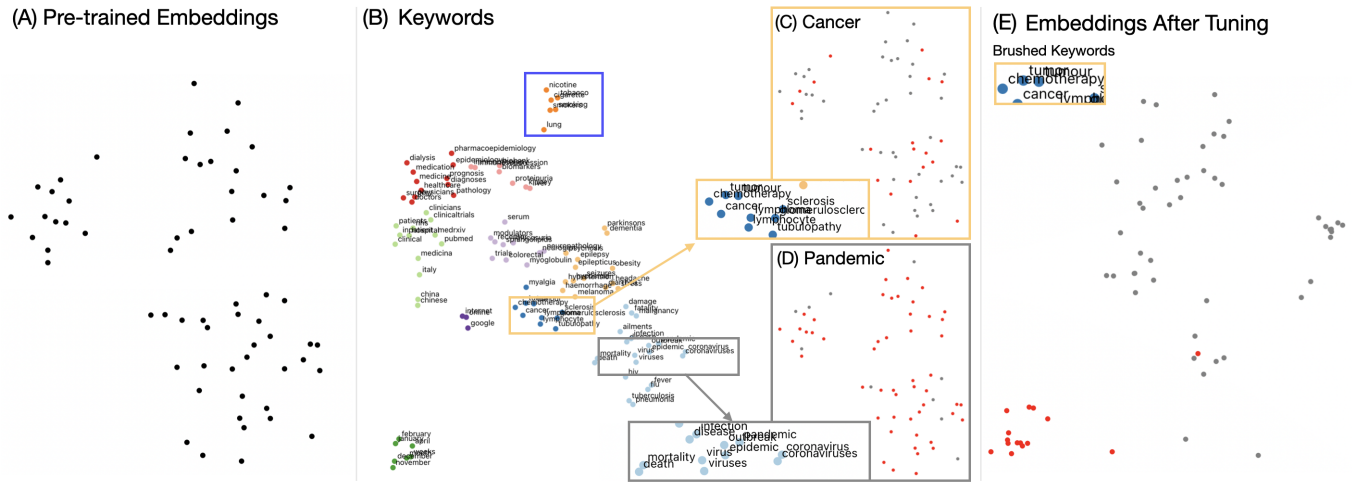

Fig. 4: An demonstration of the prototype implementation of KeySI on the COVID-19 data. (A) shows the initial pre-trained embeddings. The projection appears to contain several clusters. (B) presents a projection of the curated keywords, which can be brushed to show the linked documents. The user identifies several groups of interest - a set of words about smoking (blue box) and a set about cancer (yellow box). Brushing over the cancer keywords, they see that the cancer documents are not well organized (in (C)). They also note a set of words about the pandemic (gray box) that, upon brushing, seem to encompass the whole corpus (shown in (D)). Using the smoking and cancer keywords, they convey feedback to the model to update the embeddings. (E) shows the embeddings after tuning, with the documents corresponding to the brushed cancer keywords highlighted, demonstrating that their embeddings now capture that topic.

3.4 Model Tuning

With the pseudo-classes of documents generated by the previous step, we fine-tune the model using Triplet Margin Loss [10, 33]. We opt for Triplet Margin Loss because it aims to reduce the distance between texts in the same pseudo-class while increasing the distance from texts in different pseudo-classes in the vector space, encouraging the model to learn more semantically discriminative representations. In the remainder of this section, we provide a detailed description of the model tuning approach.

3.4.1 Triplet Margin Loss

Triplet margin loss relies on the creation of training "triplets" that hold three elements: an anchor sample, a positive sample from the same class as the anchor, and a negative sample from a different class. The anchor sample serves as the primary point for comparison while the other positive and negative provide similar and dissimilar information for training. Given a set of triplets $(\mathbf{a}, \mathbf{p}, \mathbf{n})$ where $\mathbf{a} = \{a_0 \dots a_k\}$ holds the set of anchors for the current epoch, $\mathbf{p} = \{p_0 \dots p_k\}$ holds the set of corresponding positive samples, and $\mathbf{n} = \{n_0 \dots n_k\}$ holds the negative samples, Triplet Margin Loss performs the optimization below:

$$\mathscr{L}(\mathbf{a}, \mathbf{p}, \mathbf{n}) = \sum_{i<k} \max\Big(0,\ \|f(\mathbf{a_i}) - f(\mathbf{p_i})\|^2 - \|f(\mathbf{a_i}) - f(\mathbf{n_i})\|^2 + \text{margin}\Big)$$

where $f(\cdot)$ represents the text embedding output by the BERT encoder. The hyperparameter margin controls the minimum required distance between the positive and negative samples. To balance convergence speed and discriminative power, we set margin $= 3$. By minimizing this loss, texts from the same keyword group are pulled closer in the vector space, while those from different keyword groups are pushed further apart.

3.4.2 Implementation

We implement Triplet Margin Loss using a batch cycling strategy. Rather than training on all pairs of pseudo-classes in one epoch, we cycle through individual pairs, training on only one anchor class against one other class in each epoch. To do this, we enumerate all pairs of pseudo-classes and create an iterator that cycles through each pair, returning to the start after reaching the end. Note, we compared this strategy to training on all pairs in each epoch and found the performance to be comparable, but the speed of training with batch cycling to be faster, as it does not need to consider as much information per epoch.

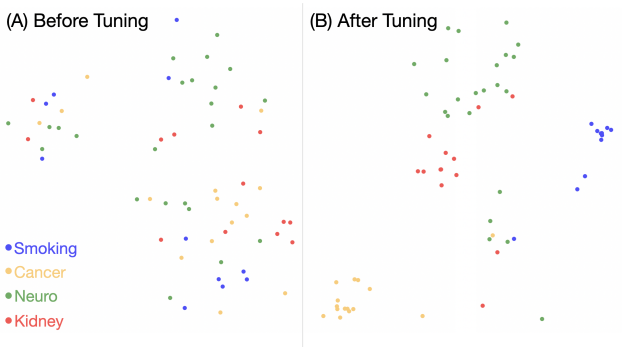

Fig. 5: For validation, the projections in Fig. 4, colored by their ground truth labels. We validate that the original embeddings fail to capture the "smoking" and "cancer" topics while the updated ones succeed. Additionally, they better recognize the remaining two topics ("neuro" and "kidney") without explicit definition.

For each epoch, given the pseudo-class pair (C_i, C_j) we construct triplets as follows. First, we draw anchor samples from C_i. We vary the number of anchors, n_a, depending on the number of documents in C_i and C_j such that it selects at least one anchor but at most $\min(0.5|C_i|, |C_j|)$. We do this to ensure that sufficient positive and negative samples remain to complete the triplets. Next, we randomly draw n_a anchors to populate the anchor vector $\mathbf{a}$. Then, we randomly draw n_a positive samples from the remaining documents in C_i to populate $\mathbf{p}$. Finally, we randomly select n_a documents from C_j to populate $\mathbf{p}$.

Optimizer and Learning Rate Scheduler. We use the Adam optimizer [19] for the BERT encoder, with an initial learning rate of 1×10^{-4}. Every 10 epochs, we halve the learning rate ($\gamma = 0.5$) to promote stable convergence and prevent overfitting.

4 Usage Scenarios

To demonstrate the utility of KeySI, we present three usage scenarios that highlight its ability to refine text embedding spaces through

user feedback. These scenarios are drawn from datasets with distinct characteristics:

- **COVID-19**: a small-scale corpus with weak semantic structure.
- **AG News**: a larger dataset with partially overlapping semantics.
- **BBC News**: a well-structured corpus with relatively clear initial clustering.

Together, these scenarios demonstrate the versatility of KeySI in adapting to diverse data conditions.

4.1 COVID-19 Risk Factors

Our first usage scenario demonstrates the ability of KeySI to convey feedback on a relatively small dataset of 62 COVID-19 abstracts, each discussing one of four risk-factors: smoking, cancer, neuro, and kidney [1]. Though the data contains four distinct risk-factors that, to a domain expert, are semantically separable, the data itself exhibits weak structure when embedded with a pre-trained model. This makes the dataset an ideal testbed for evaluating the effectiveness of KeySI in weakly structured scenarios.

Although the documents are labeled, we omit these labels during the interaction process to simulate an unlabeled setting, using them only for final validation. The scenario begins with the user loading the data into KeySI, which presents the 2D projection of document embeddings alongside the curated keyword groups, as shown in Fig. 4 (A)and (B). From the keyword projection, the user identifies several groups corresponding to potential topics, such as smoking and cancer.

Using the prototype interface, the brush over the keyword plot to identify the corresponding documents. Upon doing so, they notice a couple of things. First, the keyword groups that correspond to generic pandemic-related words Fig.4 (D), tend to occur across all documents and seem to be the overarching theme of the corpus. Second, though the initial projection appears to have some separated clusters of documents, none of the keyword clusters effectively explain those. In fact, though several of the clusters correspond to relevant topics, such as smoking and cancer, the projection did not place the corresponding documents together, Fig. 4 (C). This suggests that the embedding model did not capture these topics well.

To convey this feedback to the embedding model, the user creates two groups of keywords: one for smoking-related articles and one for cancer-related articles. They select the keywords "nicotine", "smoking", "smoker", and "cessation" to represent the "smoking" documents and "cancer", "malignancy", "tumour", "chemotherapy" to represent the "cancer" documents. Note, because their exploration revealed that general pandemic-related words relate to many documents, the user carefully avoids selecting any widely used pandemic and medical terms. Figure 4 (E) shows the projection after tuning on these topics. We see that, when the user brushes over the cancer keywords again, the projection now clusters them together.

For further validation, we color the points by their assigned labels to verify how well it discerned the "smoking" and "cancer" risk factors from the feedback groups. Figure 5 shows the projection before and after the interactive tuning, with the points colored by their ground truth label. We see that, for the most part, the feedback provided enough context to separate the "smoking" and "cancer" documents from both each other and the rest of the documents. Additionally, we see that, though the user did not define all four risk factors, the context provided was enough to better organize the two additional risk factors. KeySI allows the user to easily and intuitively express topics based on their domain knowledge to fine-tune the model to better represent those topics.

4.2 AG News Dataset

Our second usage scenario demonstrates the ability of KeySI to convey feedback on a larger dataset - AG News dataset [42]. The AG News dataset contains 2,000 documents across four high-level categories, each containing 500 documents: World, Sports, Business and Science/Technology. On the surface, the four categories seem semantically distinct, suggesting that they should be well separated in the embedding space; for example, Olympic events and international trade policy clearly belong to different semantic domains. However, upon inspecting the projection of the pre-trained embedding space (Fig. 6, left), we see that it does not effectively distinguish the four categories.

Our user begins exploring the keyword space and identifies groups of keywords that relate to large, dispersed subsets of documents in the projection, such as economy and stock. They identify keywords corresponding to the four high-level categories of articles in the dataset. To convey these topics, the user specifies the keyword groups on the right side of Fig. 6. KeySI updates the embedding space, using these keywords, to better delineate between the topics. We see from the center image of Fig. 6 that the feedback helps the model better capture the four high-level categories. From this usage scenario, we see that KeySI sufficiently captures feedback to help refine and untangle the embedding space of a larger scale dataset, with just a few keywords.

4.3 BBC news

Our final usage scenario examines the BBC News dataset which consists of 2225 documents in five predefined categories: Technology, Sports, Business, Entertainment and Politics [13]. The projection of the initial embedding space (Fig. 7, left) shows that the pre-trained model identifies these in the embedding space. However our user prefers a stronger distinction between classes, with higher separation of clusters for their downstream task.

To provide feedback, they specify the keyword groups for each of the five categories shown in Fig.7. Using these keywords, KeySI updates the embedding space. As shown in the middle image of Fig.7, the refined projection reveals clearer separations between the five categories, indicating that the keyword-based interaction effectively aligns the model with the user-defined topic semantics. This scenario demonstrates KeySI's ability to enhance the clustering structure in datasets with initially separable topics.

5 Quantitative Evaluation

To evaluate the impact of significant system settings and parameters, we conduct an empirical evaluation of the proposed KeySI method for four key settings: the variability across multiple tunings of the same feedback, the sensitivity to the precise keywords, the number of keywords necessary to effectively tune, and the impact of the percentile threshold p defined in Sec. 3.3. on three representative datasets. We evaluate all experiments using two widely adopted clustering metrics: **Silhouette Score** [29] and **Average Purity** [34] and all three datasets, described in Sec. 4.

Silhouette Score measures the quality of a clustering by calculating how similar a point is to its own cluster (cohesion) as compared to other clusters (separation). The values range from -1 to 1, where 1 indicates highly separated and tightly grouped clusters, 0 indicates overlapping clusters, and -1 indicates mis-assignments. In our setting, approximately .5 or higher indicate good clusterings - we do not prioritize overly compact clusters as they could elide semantic structures internal to a cluster. For context, the Silhouette Scores before/after interaction are 0.37/0.69 in Fig. 1, 0.40/0.48 in Fig. 6, and 0.44/0.53 in Fig. 7.

Purity measures how well the discovered clusters align with known class labels. Traditionally, Purity is computed by fixing the number of clusters k to match the number of ground-truth categories (e.g., the number of keyword groups). However, we found this approach to be limiting in practice. Rigid clustering with fixed k often overweights dominant categories and suppresses minority ones. For example, when we fix $k = 5$ on the BBC News dataset, the model may produce clusters such as sports, sports, business, business and tech, omitting the politics category entirely. This happens because K-Means prioritizes compact and well-separated clusters, which causes smaller but meaningful classes to be absorbed into larger, more frequent ones, or duplicate the most dominant category across multiple clusters.

Rather than using a fixed k, we dynamically determine k by selecting the value that maximizes the Silhouette Score, and calculating the average Purity of the resulting clustering. This allows the clustering process to adapt to natural topic granularity and better represent the underlying semantic structure. We empirically compared both strategies and found

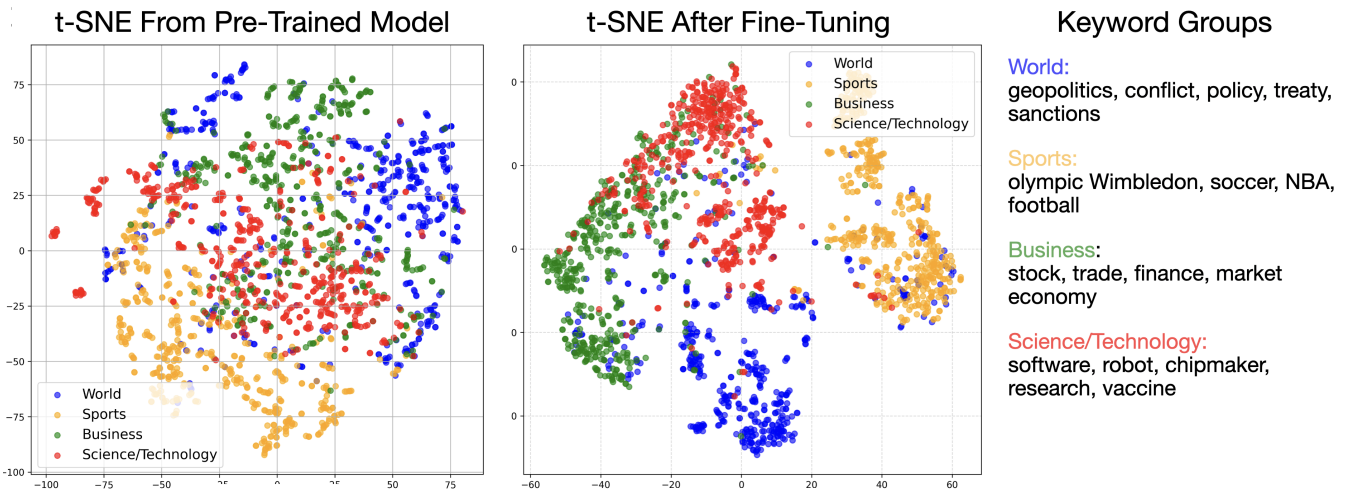

Fig. 6: The left image shows the projection of the AG news dataset using pre-trained embeddings. Though the topics appear semantically distinct, the pre-trained embeddings do not sufficiently identify them. By defining relevant keyword groups, the user expresses the topics to KeySI which incorporates this feedback into the tuned model. The center image shows the projection of the tuned model after interaction where we see better definition and separation of the documents in each topic.

that dynamically selecting k yields higher-quality clusters and better preserves topic diversity — particularly in datasets with complex internal structure. In contrast, fixing k often results in overlooked minority clusters and over-generalized topic boundaries.

This flexible clustering evaluation strategy enables KeySI to more accurately reflect real semantic distributions and more reliably assess the effectiveness of user-driven feedback. Further discussion of these trade-offs can be found in Section 6.

5.1 Stability Under Repeated Interactions

To assess the stability of KeySI, we repeated the same interaction process 30 times on the three representative datasets. For each run, we recorded the Average Purity and Silhouette Score of the resulting document embeddings.

As shown in Fig. 8 (A), the results demonstrate overall stability across runs, with high and consistent Average Purity scores on all datasets. Both BBC News and COVID-19 datasets demonstrated high average purity, with AG News following closely behind. While some outliers are observed—particularly in Average Purity—the interquartile ranges remain narrow, indicating the system performs reliably in most cases. Silhouette Scores also show limited variance, further supporting the consistency of the structural separation achieved through KeySI. These results suggest that, despite occasional variability, KeySI consistently produces meaningful semantic clustering under repeated executions with the same feedback.

5.2 Robustness to Keyword Variations

Our second test aims to evaluate the robustness of the system to variations in the specific keywords used to define the topic. For this test, we generate five semantically equivalent variations of each keyword group and evaluate how consistently the resulting embeddings preserve topic structure across these variants. Note, due to the small sample size of the COVID-19 dataset, we were unable to generate 5 meaningful variations of the topics and thus only perform this evaluation on the larger news datasets.

According to Table. 1, KeySI demonstrates strong robustness across five semantically equivalent keyword group variations, even when less representative or potentially noisy terms are included. The resulting embeddings exhibit less than 9% deviation in both Silhouette Score and Average Purity, indicating stable performance under varied user input.

Dataset	Silhouette Score	Average Purity
AG-1	0.4827	0.7810
AG-2	0.4512	0.6790
AG-3	0.4470	0.7882
AG-4	0.4797	0.8084
AG-5	0.5050	0.7799
BBC-1	0.5304	0.8004
BBC-2	0.5024	0.8100
BBC-3	0.5252	0.8739
BBC-4	0.5250	0.8006
BBC-5	0.4571	0.7169

Table 1: Silhouette Score and Average Purity for Different Keyword Groupings (AG news and BBC news)

5.3 Effect of Varying Number of Keywords

Next, we evaluate how well KeySI captures feedback when using smaller versus larger keyword groups. To do this, we vary the number of keywords in each group from one to five and measure the performance of KeySI by evaluating the clustering in each resulting embedding.

The results, shown in Fig. 8 (B), demonstrate that, for the COVID-19 dataset, performance improves consistently up to 4 keywords, achieving the best purity and Silhouette Score. We observe a slight drop at five keywords, due to the inclusion of less semantically significant terms(e.g.,lung).Although that keyword brings in documents related to smoking, it also introduces some with only weak semantic relevance. These borderline cases are difficult to eliminate even with distance-based thresholding, as they fall within the acceptable range but still dilute the coherence of the topic. This indicates the need for precision in keyword selection, especially in smaller datasets.

In the AG News dataset, increasing keywords per group consistently improves clustering quality. Unlike COVID-19 dataset, performance continues to rise up to five keywords without degradation (Fig. 8 (B). This suggests that in larger datasets, higher keyword counts improve semantic coverage and clustering effectiveness.

Similarly, in the BBC News data, the clustering purity remains relatively high and stable throughout (Fig. 8 (B)). This shows KeySI achieves effective clustering with minimal user input on well-structured datasets.

These results demonstrate that KeySI effectively captures user feedback with varying keyword group sizes. While larger keyword sets

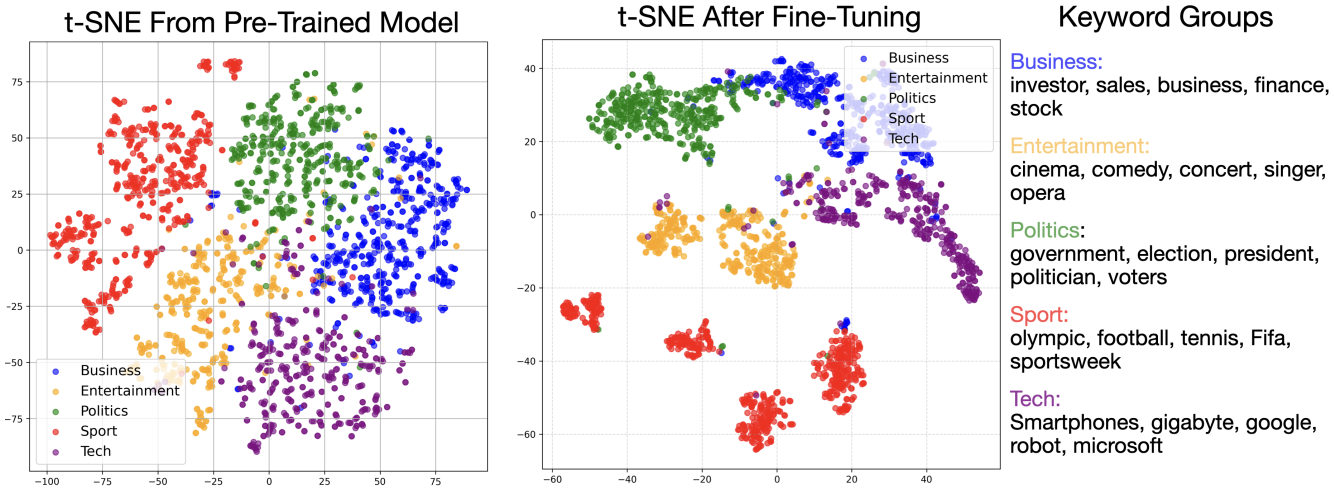

Fig. 7: The left image shows the projection of the BBC news dataset using pre-trained embeddings. Though the pre-trained embeddings identify the overarching topics, KeySI helps better refine and separate those topics, by specifying the keyword groups on the right. The center image shows the projection of the tuned model after interaction.

improve performance in large datasets by enhancing semantic coverage, smaller datasets benefit more from precise, carefully chosen keywords. This suggests that optimal keyword group size should be adapted to the scale and complexity of the dataset.

5.4 Threshold Sensitivity

Finally, we evaluate the sensitivity of the training to the percentile threshold (p) defined in Sec. 3.3. Recall that p determines the threshold for including or excluding candidate documents from the generated pseudo-classes. Specifically, when looking at the distribution of distances of documents to the cluster centroid, KeySI preserves documents that are below the p-th percentile and eliminates those outside of it. In this test, we examine the impact of different choices of p and validate the default choice included in KeySI. For each dataset, we use the same keyword groups but vary the value of p from 5% to 40%, at increments of five.

As shown in Fig. 8 (C), the Silhouette score and Average Purity both peak at the 25% threshold for the COVID-19 and AG News data, indicating an optimal balance between semantic proximity and document coverage. BBC News performs comparably, though with its highest performance occurring around 15% based on the silhouette and 35% based on purity, with a slight dip in Silhouette score at 25%. However, the BBC news still exhibits good performance at 25% and thus we choose this setting for subsequent experiments. We also observe that, for the small dataset, increasing the threshold beyond 25% results in a decrease in performance. Intuitively, this makes sense as, with fewer documents in each topic to draw from, a larger threshold increases the chances of drawing in less relevant documents. This test validates that a lower threshold is reasonable for both small and large datasets.

5.5 Summary

Across all three datasets, KeySI demonstrates strong and consistent performance in enhancing clustering quality through keyword-driven semantic interactions. The method exhibits robustness across varying dataset sizes, structural characteristics, and keyword configurations.

Compared to the baseline performance (Table 2), KeySI achieves substantial improvements on datasets with initially weak semantic structure, such as COVID-19 and AG News, significantly enhancing both the coherence and separability of clusters. For the BBC News dataset, where the baseline clustering is already well-structured, KeySI maintains comparable Average Purity while notably improving the Silhouette Score. This suggests that the refined embedding space provides clearer boundary definitions and facilitates better topic distinction, even in high-quality baselines.

Dataset	Silhouette Score	Average Purity
COVID-19 Risk Factors	0.3693	0.33
AG news	0.3977	0.65
BBC news	0.4411	0.86

Table 2: Baseline clustering performance on three datasets. labels.

6 Discussion

Comparison to DeepSI KeySI and DeepSI exhibit fundamentally different design philosophies in terms of interaction mechanisms. DeepSI relies on manual manipulations of documents in a 2D space—such as dragging documents to express semantic relationships—which requires users to have a strong understanding of the document content or the structure of the embedding space. In contrast, KeySI focuses on interaction at the keyword level, allowing users to express semantic intent by organizing groups of related keywords, without needing prior knowledge of the embedding space distribution. In essence, KeySI uses keywords as a preliminary step before applying a DeepSI-style pipeline - the keywords specify the relevant context and then KeySI automatically identifies the relevant documents to feed into the DeepSI-style pipeline. This eliminates the need for users to manually inspect documents to convey feedback, providing a more user-friendly experience, especially in tasks involving large document collections or lacking labeled data.

Generalizability of Interaction Pipeline In our current implementation, we adopt KeyBERT for keyword extraction, use MDS for dimensionality reduction, and apply a percentile-based thresholding strategy to filter out semantically marginal documents after matching them based on keyword groups. t-SNE is primarily used to visualize the overall structure of the embedding space before and after model tuning, helping users qualitatively assess the effectiveness of the refinement process, while BERT serves as the language model for generating semantic embeddings. Considering the diversity of data characteristics, task goals, and user preferences, we deliberately designed a flexible interaction paradigm that allows users to customize the methods for keyword extraction, dimensionality reduction, and embedding according to their specific needs.

Granularity of Keyword Feedback Though the results in this paper show promise, keywords may sometimes be too fine-grained to provide concise feedback. In particular, as datasets grow in scale, the number of keywords selected will increase substantially. Even with the mechanisms to guide users to commonly used words and to

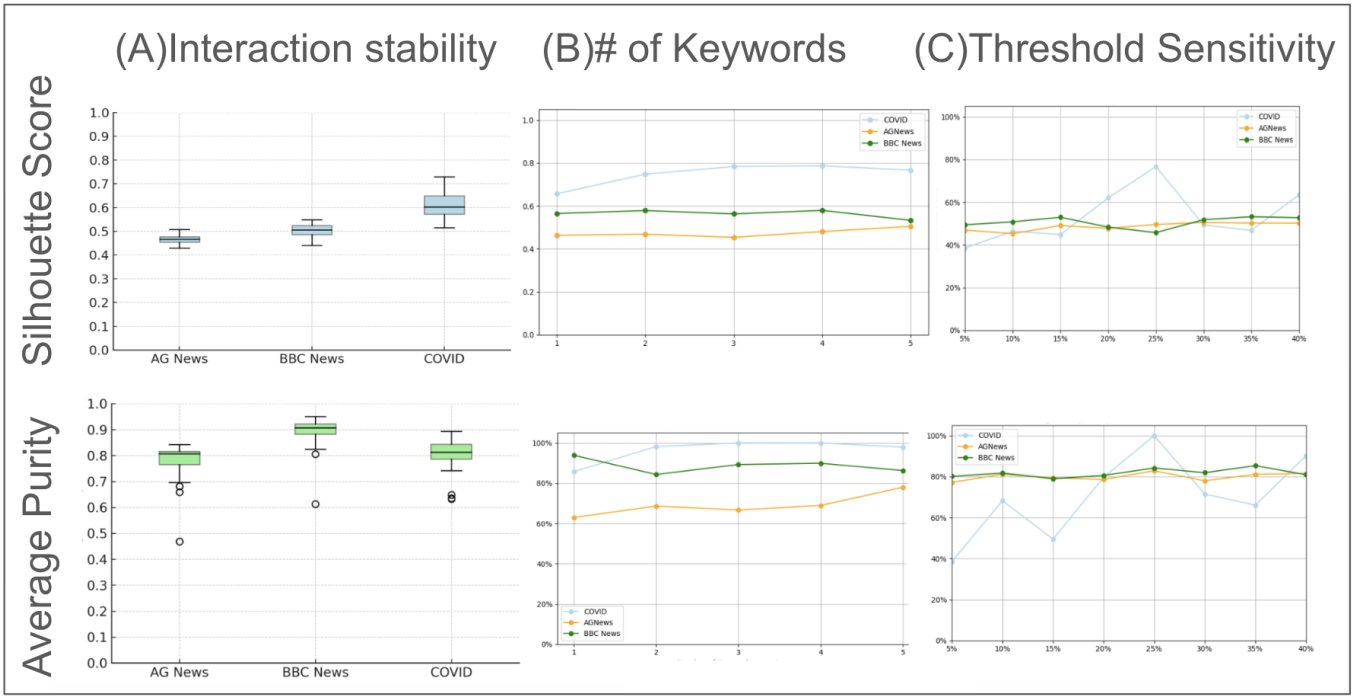

Fig. 8: The Silhouette Score (top row) and average purity (bottom row) across three of our tests: (A) interaction stability, (B) varying number of keywords, and (C) threshold sensitivity. Sec. 5 describes each of these tests.

aggregate similar categories of words to reduce the visual search space provided by KeySI, the final list of keywords may still be quite long. To overcome this, future directions can investigate the use of topic modeling to provide more concise descriptions of the text corpus, as well as plain text descriptions of concepts that can be compared against individual documents to assess similarity.

Keyword Selection Consideration Our study found that over-general or ambiguous keywords may diminish clustering quality. For example, broad keywords such as "disease" tend to be associated with multiple, distinct topics inside different document categories, thus leading to ineffective and ambiguous clustering results during our evaluation. At the same time, specific terms, although intuitively relevant, might still introduce unwanted noise. For example, we found that the term "lung" frequently occurs in both smoking-related and cancer-related articles, which leads to the evaluation of semantic clarity of intended keyword groups becoming insufficient. To address this limit, users should avoid keywords that are overly broad or ambiguous. Besides, we also advise users to select specific, well-defined terms strongly associated with their intended topic categories—such as "chemotherapy," which reliably indicates cancer-related content—to enhance clustering clarity and model performance.

Limitations in Cluster Evaluation Our evaluation method has inherent limitations, as typical cluster evaluation methods rely heavily on manually assigned labels, often oversimplifying the underlying semantic complexity of documents. Typically, documents may contain multiple semantic dimensions simultaneously, so a single label is often insufficient to capture the subtle semantic context accurately. For example, documents labeled as “business” often contain much discussion related to technology, innovation, or software topics, making it difficult to categorize them under a single topic. To mitigate this limitation, we tried to validate our clustering by combining multiple clustering evaluation metrics (e.g., the Silhouette Score and purity). However, these metrics still cannot fully represent the richness of semantics unless the documents are manually inspected. Future work may explore more complex semantic perception evaluation methods to reflect better the true complexity and multidimensionality of the text corpus.

Interactions & Loss Functions KeySI opts for a cluster-focused interaction, through the definition of disjoint keyword groups and the application of Triplet Margin Loss, a cluster focused loss function. Though this provides an intuitive means for conveying feedback, it may not encompass all user tasks. For instance, it does not support defining inter-topic relationships beyond very discrete clustering. However, some tasks may be better supported through more complex relationships - e.g., through the definition of a hierarchy of topics or explicit cluster similarities. Future work should explore designing intuitive interactions corresponding loss functions for more complex feedback

Connecting Document & Text Feature Spaces Finally, we recognize a fully effective system requires explainability in the document projection to connect information in the text feature space to the current document embeddings. Our prototype provides light-weight interactions to assist in connecting the two spaces, but future work is needed to provide richer context and deeper connection between the document and text feature spaces to better guid interactions.

7 Conclusion

This paper presents KeySI, an interactive semantic embedding optimization framework centered around keyword-level feedback. Unlike most existing approaches that rely on manual labels or document-level interactions, KeySI focuses on user-defined keyword groups, eliminating the need for prior annotations and significantly reducing user interaction overhead. This design enhances the scalability and practicality of the method, especially in large-scale, unlabeled corpora.

KeySI automatically constructs pseudo-labels from user-defined keyword groups and performs unsupervised optimization of the embedding space using Triplet Margin Loss. During this process, we introduce a semantic filtering mechanism that applies clustering and distance-based thresholds to select a more representative subset of documents from the candidates, thereby improving the quality of pseudo-labels and the effectiveness of model training.

Experimental results demonstrate that KeySI exhibits strong stability and robustness across multiple datasets. It adapts well to varying numbers and configurations of keyword inputs, effectively enhancing the semantic clarity and clustering quality of the embedding space.

In future work, we aim to support more flexible and higher-level forms of user feedback and further refine the design of loss functions to improve the model's ability to capture complex semantic relationships and enhance generalization.

REFERENCES

[1] A. I. F. AI, A. Goldbloom, B. Hamner, C. Schoenick, T. Bozsolik, P. Mooney, P. Lin, S. Kohlmeier, and devrishi. COVID-19 Open Research Dataset Challenge (CORD-19), 2022. 6

[2] R. Bekkerman, H. Raghavan, J. Allan, and K. Eguchi. Interactive clustering of text collections according to a user-specified criterion. In *IJCAI*, vol. 7, pp. 684–689, 2007. 2

[3] Y. Bian, R. Faust, and C. North. Neuralsi: Neural design of semantic interaction for interactive deep learning. *arXiv preprint arXiv:2402.17178*, 2024. 1, 3

[4] Y. Bian and C. North. Deepsi: Interactive deep learning for semantic interaction. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, pp. 197–207, 2021. 1, 3

[5] Y. Bian, J. Wenskovitch, and C. North. Deepva: Bridging cognition and computation through semantic interaction and deep learning. In *2019 IEEE Workshop on Machine Learning from User Interaction for Visualization and Analytics (MLUI)*, pp. 1–10. IEEE, 2019. 3

[6] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly, Beijing Cambridge [Mass.], 2009. 3

[7] L. Bradel, N. Wycoff, L. House, and C. North. Big text visual analytics in sensemaking. In *2015 Big Data Visual Analytics (BDVA)*, pp. 1–8. IEEE, 2015. 2

[8] M. Dowling, N. Wycoff, B. Mayer, J. Wenskovitch, L. House, N. Polys, C. North, and P. Hauck. Interactive visual analytics for sensemaking with big text. *Big Data Research*, 16:49–58, 2019. 2

[9] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 473–482, 2012. 3

[10] A. Feragen, F. Lauze, and S. Hauberg. On geodesic exponential kernels. In *3rd International Workshop on Similarity-Based Pattern Recognition (SIMBAD 2015)*, pp. 211–213. Springer, 2015. 3, 5

[11] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *IEEE transactions on Visualization and Computer Graphics*, 19(10):1646–1663, 2012. 2

[12] M. Grootendorst. Keybert: Minimal keyword extraction with bert., 2020. doi: 10.5281/zenodo.4461265 3

[13] H. Gültekin. BBC News Archive, 2020. 6

[14] X. Ho, A. K. D. Nguyen, A. T. Dao, J. Jiang, Y. Chida, K. Sugimoto, H. Q. To, F. Boudin, and A. Aizawa. A survey of pre-trained language models for processing scientific text, 2024. 1

[15] Y. Hu, E. E. Milios, and J. Blustein. Interactive feature selection for document clustering. In *Proceedings of the 2011 ACM symposium on applied computing*, pp. 1143–1150, 2011. 2

[16] Y. Hu, E. E. Milios, and J. Blustein. Interactive document clustering with feature supervision through reweighting. *Intelligent Data Analysis*, 18(4):561–581, 2014. 2

[17] Y.-a. Kang and J. Stasko. Examining the use of a visual analytics system for sensemaking tasks: Case studies with domain experts. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2869–2878, 2012. 2

[18] S. Kiefer, M. Hoffmann, and U. Schmid. Semantic interactive learning for text classification: a constructive approach for contextual interactions. *Machine Learning and Knowledge Extraction*, 4(4):994–1010, 2022. 2

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. 5

[20] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer graphics forum*, vol. 31, pp. 1155–1164. Wiley Online Library, 2012. 2

[21] J. Lin, R. Faust, and C. North. Imagesi: Semantic interaction for deep learning image projections. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 91–95. IEEE, 2024. 1, 3

[22] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 543–552, 2009. 2

[23] E. Loper and S. Bird. NLTK: the natural language toolkit. *CoRR*, cs.CL/0205028, 2002. 3

[24] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. 4

[25] S. Mishra and J. M. Rzeszotarski. Designing interactive transfer learning tools for ml non-experts. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021. 1, 2

[26] S. Nourashrafeddin, E. Milios, and D. Arnold. Interactive text document clustering using feature labeling. In *Proceedings of the 2013 ACM symposium on Document Engineering*, pp. 61–70, 2013. 2

[27] S. Rezaeipourfarsangi, N. Pei, E. Sherkat, and E. Milios. Interactive clustering and high-recall information retrieval using language models. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, pp. 1–5, 2022. 2

[28] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 4

[29] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 4, 6

[30] T. Ruppert, M. Staab, A. Bannach, H. Lücke-Tieke, J. Bernard, A. Kuijper, and J. Kohlhammer. Visual interactive creation and validation of text clustering workflows to explore document collections. *Electronic Imaging*, 29:46–57, 2017. 2

[31] H. Sakai and S. Masuyama. A multiple-document summarization system with user interaction. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1001–1007, 2004. 2

[32] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015. 3, 5

[34] H. Schütze, C. D. Manning, and P. Raghavan. *Introduction to information retrieval*, vol. 39. Cambridge University Press Cambridge, 2008. 6

[35] P. Sen, Y. Li, E. Kandogan, Y. Yang, and W. Lasecki. Heidl: Learning linguistic expressions with deep learning and human-in-the-loop. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 135–140, 2019. 2

[36] E. Sherkat, E. E. Milios, and R. Minghim. A visual analytics approach for interactive document clustering. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 10(1):1–33, 2019. 2

[37] E. Sherkat, S. Nourashrafeddin, E. E. Milios, and R. Minghim. Interactive document clustering revisited: A visual analytics approach. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pp. 281–292, 2018. 2

[38] J. Stasko, C. Gorg, Z. Liu, and K. Singhal. Jigsaw: supporting investigative analysis through interactive visualization. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 131–138. IEEE, 2007. 2

[39] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 3, 4

[40] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun. Pre-trained language models and their applications. *Engineering*, 25:51–65, 2023. 1

[41] Y. Yang, E. Kandogan, Y. Li, P. Sen, and W. S. Lasecki. A study on interaction in human-in-the-loop machine learning for text analytics. In *IUI Workshops*, 2019. 2

[42] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015. 6