Intelligent and Converged Networks ISSN 2708-6240
2024, 5(1): 42−52 DOI: 10.23919/ICN.2024.0004

RecBERT: Semantic recommendation engine with large language model enhanced query segmentation for *k*-nearest neighbors ranking retrieval

Richard Wu*

Abstract: The increasing amount of user traffic on Internet discussion forums has led to a huge amount of unstructured natural language data in the form of user comments. Most modern recommendation systems rely on manual tagging, relying on administrators to label the features of a class, or story, which a user comment corresponds to. Another common approach is to use pre-trained word embeddings to compare class descriptions for textual similarity, then use a distance metric such as cosine similarity or Euclidean distance to find top *k* neighbors. However, neither approach is able to fully utilize this user-generated unstructured natural language data, reducing the scope of these recommendation systems. This paper studies the application of domain adaptation on a transformer for the set of user comments to be indexed, and the use of simple contrastive learning for the sentence transformer fine-tuning process to generate meaningful semantic embeddings for the various user comments that apply to each class. In order to match a query containing content from multiple user comments belonging to the same class, the construction of a subquery channel for computing class-level similarity is proposed. This channel uses query segmentation of the aggregate query into subqueries, performing *k*-nearest neighbors (KNN) search on each individual subquery. RecBERT achieves state-of-the-art performance, outperforming other state-of-the-art models in accuracy, precision, recall, and F1 score for classifying comments between four and eight classes, respectively. RecBERT outperforms the most precise state-of-the-art model (distilRoBERTa) in precision by 6.97% for matching comments between eight classes.

Key words: sentence transformer; simple contrastive learning; large language models; query segmentation; *k*-nearest neighbors

1 Introduction

1.1 Background

Finding meaningful representations of text documents to compute document-level similarity has been a classic problem in the field of natural language processing (NLP). In 1954, the bag of words (BoW) model was proposed by Harris[1], creating a vector representation for each document composed of the frequencies of each word in a govern vocabulary. BoW models have been enhanced with the addition of term frequency-inverse document frequency (TF-IDF)[2]. However, BoW models are not scalable for larger sets of documents, as the dimensions of sparse frequency vectors will grow extremely large as the vocabulary increases for larger text corpuses[3]. Conventional distance metrics such as Euclidean distance become significantly less effective for computing similarity in high dimensional vectors, as the distances between nearest and farthest vectors become nearly indistinguishable from each other[3].

Recent progress with the transformer architecture proposed in 2017 by Vaswani et al.[4] introduces parallelizable computations while also maintaining core attention mechanisms for contextualized word

• Richard Wu is with the Dublin Unified School District and the SF Artificial Intelligence Club, Dublin, CA 94568, USA. E-mail: kingdomcalifornia@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2023-09-18; revised: 2023-09-30; accepted: 2023-10-10

embeddings. It uses dense vector embeddings to represent documents as opposed to the sparse frequency vectors generated from BoW models. BERT[5] uses masked language modeling (MLM) as opposed to casual language modeling, which allows it to generate bidirectional representations of words[5].

1.2 Motivation

With the increasing number of user-generated recommendations and descriptions for documents in discussion forums, the amount of unstructured natural language data has greatly increased. However, most modern recommendation systems use manual tagging of a document's features to perform collaborative[6–10] or content-based[11–13] filtering. Some directly use pre-trained word embeddings to compare queries with descriptions for textual similarity, then use a distance metric such as cosine similarity or Euclidean distance to find top k neighbors. These approaches do not utilize the full potential of user-generated unstructured natural language data in their recommendations.

Most semantic embedding models such as Word2Vec or BERT are trained on large-scale general-domain datasets, such as Wikipedia. While these pret-rained representations have a good understanding of general language patterns, their performance may degrade for more specific domains, such as user comments for stories. That is, the context necessary for the transformer to learn on the domain of general articles compared to the domain of user comments may differ.

Additionally, there is the issue of user queries potentially containing content from multiple sources. Finding textual similarity between the full query and the individual documents (user comments), each containing a fragment of the user query, is not ideal in finding accurate similarity across classes. Figure 1 shows that when a query is essentially the combination of two separate documents, both corresponding to the same class, the similarity between the query and each document individually is lower than its true similarity when considering the query directly against the class. A k-nearest neighbors (KNN) search based only on the full query's embedding is likely to return erroneous

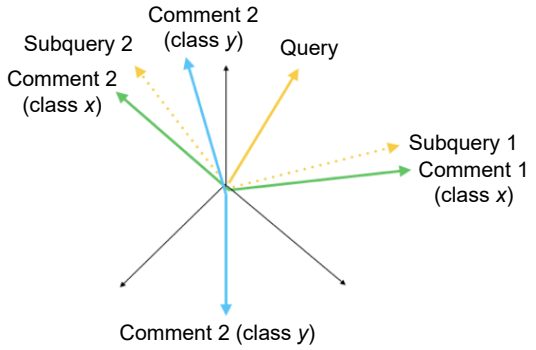


Fig. 1 Visualization of splitting a query into its subqueries for improved accuracy.

results when requiring components from different comments belonging to the same class in order to separately match to each subquery. For example, the class to be ranked, class x, may contain a comment corresponding to subquery 1, and another comment corresponding to subquery n. Directly comparing semantic similarity between the aggregate query and each comment containing a component of that query won't fully encapsulate the similarity between the class and query. Thus, in order to properly match the query against the documents, or comments, corresponding to a class, the query must be split into its subqueries and have KNN search independently conducted on each subquery.

1.3 Main idea

This paper proposes a semantic recommendation system, RecBERT, by first further pre-training a bidirectional transformer using MLM[5] for domain adaptation on all user comments for the set of classes. Distance metrics such as cosine similarity can be used to rank comments most similar to the query. Without the initial domain adaptation, the fine tuning of the sentence transformer afterward will yield very poor results, overfitting on specific phrases, limiting the scope of top k neighbor results. In this paper, domain adaptation will be applied to story recommendations by adapting on a dataset of user-generated review comments, but this approach can also be used for other domains, such as biology texts[14] or medical diagnosis by collecting patient reports (which are the equivalent of user comments in this scenario) for varying diseases[15].

It is important to note that directly averaging the pooled domain adapted BERT embeddings yields very poor results[16, 17]. This leads to the need to fine-tune the domain adapted model using a Siamese network[18] with simple contrastive learning[19] (SimCSE) to produce a model capable of encoding accurate semantic representations of user comments. Additionally, for finding comment similarity, a cross-encoder structure would be viable for training, but during runtime, would be too computationally expensive to conduct sentence similarity across large datasets[18]. Thus, a Siamese network is used to compute absolute vector representations of texts, removing the need to run each text pair through a classification network to compute similarity. With absolute sentence-level embeddings, cosine similarity or another distance metric can be used directly to conduct a similarity search such as KNN or approximate nearest neighbors on comment and query embeddings to retrieve similar user comments to the query.

To solve the task for query segmentation, we propose using a pre-trained large language model (LLM) with few-shot prompting to segment each user query into relevant subqueries to conduct cosine similarity on with the user comments available. The few-shot samples would preferably come from the same domain as the user comments for maximal accuracy in segmentation. By finding cosine similarity between subqueries and the available user comments, we are able to consider all the components of a user query to holistically determine the most similar class. The original query will also have cosine similarity performed against user comments, and its top k neighbors found, the justification being that direct matches between the aggregate query and comment are still possible, and should be weighted more than a subquery match. We conducted the experiment on the myanimelist dataset, and achieved state-of-the-art performance.

Overall, the main contributions of this study are:

(1) RecBERT is the first BERT based model further pre-trained on the MyAnimeList dataset of user comments and fine-tuned for comment embeddings with SimCSE to construct a recommendation system.

(2) Experimental results on the comment dataset demonstrate that further pre-training BERT on user comments improves its performance. RecBERT obtains state-of-the-art performance on determining the class a comment corresponds to.

(3) Study the application of LLM-enhanced query segmentation for effective KNN search on most similar classes to the query.

(4) Propose a recommendation system using LLM-enhanced query segmentation to construct a separate subquery channel for computing class level similarity to enhance ranking retrieval performance.

2 Related work

2.1 Recommendation systems

Many recommendation systems have been proposed with various architectures. Some prominent methods include collaborative[6–10] and content-based[11–13] filtering. Content filtering refers to the idea that users should be recommended items similar to those they viewed or purchased before[11]. This is a useful method for recommending products in a website the user has frequently purchased from, but in the case where they have not had any traffic on the website, it is unviable. The assumption is that all products, or classes, must have labels assigned to them showing the features they contain[11]. The prerequisite for content-based filtering is data labeled with features. A similar requirement exists with collaborative filtering, which assumes that users who shared a similar purchase or interest will have other similar interests as well[6]. In addition, if the user wants to send a query in natural language form, collaborative or content filtering are infeasible. RecBERT aims to mitigate this problem by directly finding most similar comments to the query through conducting cosine similarity on their semantic embeddings.

Another type of recommendation system adds a final layer to a BERT model for downstream classification[20, 21]. This architecture allows for users to give text input and expect a list of probabilities of the classes it most likely corresponds to, which makes

it more flexible than content-based or collaborative filtering[7]. However, the lack of document-level embeddings makes it difficult to ascertain the exact document that led to these probabilities. RecBERT conducts KNN search on comment embeddings, adding transparency as to what comments in the top returned classes were the cause of the class's high similarity ranking.

Another type of recommendation system, specifically for Q&A, uses KNN search[22] to find the most probable responses. This is performed at the contextualized word embedding level of BERT, while RecBERT works at the sentence, or document level, fine-tuning BERT using SimCSE with margin ranking loss (MNR). RecBERT also adds query segmentation to user queries as a separate channel to find most similar classes instead of only conducting KNN search on the full user query.

2.2 Domain adaptation

Domain adaptation on the base BERT model has been applied in many fields, such as biological texts[14] and medical texts[15]. The authors of BioBERT describe the success of domain adaptation for biological texts, noting increases in F1 score and MNR improvements[14]. MedBERT's domain adaptation has also improved the AUC for producing accurate contextualized embeddings for electronic health records (EHR)[15]. However, domain adaptation in these cases is being applied to produce contextualized word embeddings. These models have not been applied for generating sentence-level embeddings. RecBERT uses domain adaptation for user comments on stories, specifically the myanimelist dataset. It uses domain adaptation to help create more accurate contextualized embeddings for the user comments, while also preventing overfitting once applying sentence-level fine-tuning. RecBERT then uses the domain adapted transformer model for sentence-level embeddings by fine-tuning it with SimCSE with MNR.

2.3 LLM with few-shot learning

The method of using LLMs with few-shot learning[23] to automate previously human-done tasks has been applied to a variety of different fields, such as reinforcement learning through AI feedback[24] (RLAF) and Q&A. Since the popularization of open source LLMs after LLaMa[25], there have been many instruction fine-tuned models created with LLaMa as the base model. The introduction of quantized language representation adaptation (QLoRA)[26] has further enhanced the capabilities to fine-tune quantized LLMs. It has made it more feasible to fine-tune LLMs even on consumer grade GPUs with limited computational resources[26].

The usage of local LLMs with few-shot learning[23] makes it possible to run RecBERT, specifically query segmentation, locally on consumer grade hardware. RecBERT utilizes LLM-enhanced query segmentation to conduct cosine similarity between all comments and each subquery, creating a separate channel for computing class level similarity to subqueries, enhancing ranking performance.

3 Method

The framework of RecBERT consists of three key components. RecBERT first performs domain adaptation, or further pre-training, on the dataset consisting of user comments. Then, the domain adapted transformer is fine-tuned for document-level similarity using SimCSE, a method useful for fine-tuning on unlabeled data such as user comments[19]. Query Segmentation enhanced with LLMs is used to conduct KNN search independently on all subqueries, using a list of similarities for each subquery per class for further ranking retrieval.

3.1 Domain adaptation of BERT

The base transformer architecture of RecBERT is the same as BERT. BERT was pre-trained on general-domain datasets, namely Wikipedia and the Brown Corpus. The domain of user comments for stories has various domain-specific language (e.g., Isekai, Slice of Life, Deus Ex Machina) understood by readers. This causes the classic BERT model to perform poorly on generating meaningful semantic representations of user comments. Figure 2 shows how the domain adaptation of the BERT model to the user comments dataset allows better contextualized representations of the

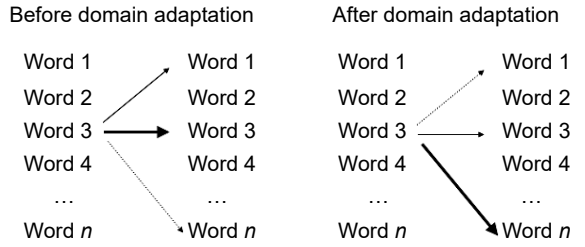


Fig. 2 Example of domain adaptation improving accuracy of contextualized word embeddings.

vocabulary.

First, a dataset consisting of classes and the user comments corresponding to them is acquired. The RoBERTa model is used as the base model for further pre-training. All user comments are tokenized using the same tokenizer as the transformer base model. The masking probability to conduct MLM with is 15%, the same as what was proposed in the original BERT paper. Further pre-training using MLM is conducted on the user comments for 10 epochs for domain adaptation.

3.2 Fine-tuning domain adapted BERT with Siamese network and SimCSE

Directly averaging the pooled domain adapted BERT embeddings yields very poor results. To compute comment-level similarity instead of word-level similarity, Siamese network[18] with SimCSE [19] and MNR is proposed to produce a model capable of encoding accurate semantic representations of user comments. Absolute comment-level embeddings are preferred over the cross-encoder structure due to far less computation being needed during query time. SimCSE is preferred over other training methods due to removing the need for labeled data while fine-tuning, which removes the need to make the assumption that all comments related to a class must be similar. It is critical to not assume this, as there can be multiple titles, or classes, that share similar themes, and thus have comment overlap. The usage of SimCSE for fine-tuning comment embeddings is as follows:

Let α_i be the current comment to conduct MNR on.

For α_i, a batch of X comments may be defined as

$$P = A$$

$$N = \{\forall n \subseteq X | n \neq A\}$$

$$|A| = |P| = 1$$

$$|N| = |X| - 1$$

with A equal to α_i as the anchor, P equal to A as the positive, and N as the set of all other comments within X.

Cosine similarity used for similarity comparison between comment embeddings, defined as

$$D(A,B) = \frac{|\boldsymbol{A} \cdot \boldsymbol{B}|}{\|\boldsymbol{A}\| \times \|\boldsymbol{B}\|} \tag{1}$$

Let $e(x)$ represent the embedding function to convert a text object x into its sentence embedding. The multiple negatives ranking loss function for batch X can be defined as follows:

$$L(A,R,N) = \sum_{i=1}^{n} (D(e(A), e(P)) + D(e(A), e(N_i)))$$

Because $P = A$, this can be simplified to

$$L(A,P,N) = \sum_{i=1}^{n} (D(e(A), e(A)) + D(e(A), e(N_i))) \tag{2}$$

First, the saved domain adapted transformer is used as the base for fine-tuning. The same dataset consisting of story titles (class labels) and the user reviews corresponding to them is used as training data for fine-tuning at the comment level. Sentence pairs of the same sentence are generated, which will have different encodings when passed into the transformer model due to dropout[19]. A batch size, such as 128, is used for training. Too large of a batch size may prevent convergence. The multiple negatives ranking loss function is used during training to minimize distances between the same sentence, and maximize distances between all other sentences in the batch. The domain adapted transformer is fine-tuned for 20+ epochs.

3.3 Query segmentation with LLM for ranking retrieval

3.3.1 Query segmentation with LLM

While comment embeddings are now more accurately represented, there is still the issue of the full user query potentially containing content from multiple sources in the same class. This may cause inaccurate search results, as using cosine similarity between all comments and the full query will give lower

similarities between the full query and a comment consisting of one section of that query, while not taking into account the other comments belonging to the same class that constitute the additional sections of that query. That is, the similarity between the query and each comment individually is lower than its true similarity when considering the query directly against the class. To solve this problem, query segmentation with an LLM is proposed.

Let γ represent the user query, Assume $\exists$ N subqueries $\{\gamma_1, \gamma_2, ..., \gamma_N\} \subseteq \gamma$.

A pre-trained LLM assistant model performs the segmentation task $F(\gamma) = \{\gamma_1, \gamma_2, ..., \gamma_N\}$, segmenting an aggregate query into its subqueries.

The assistant model is given a prompt to segment the query into its parts, and is provided few-shot samples of the segmentation task.

Format of prompt and responses:

Consider the following user query:

[Instructions to perform segmentation on the separate themes of the query]

Query: [Example User Query]

Segmented Query:

[Example Segment 1]

[Example Segment 2]

[Example Segment *n*]

Query: [Actual User Query]

Segmented Query:

[To be filled in by assistant model]

The returned segmented output $\{\gamma_1, \gamma_2, ..., \gamma_N\}$ may then be used for ranking retrieval.

3.3.2 Ranking retrieval

To find the classes that best match the query, segment the original query γ into $\{\gamma_1, \gamma_2, ..., \gamma_N\}$. Let Q be defined as $\{\gamma_1, \gamma_2, ..., \gamma_N\}$. KNN search is performed on each element in Q, outputting the most similar comment c_i corresponding to each subquery for each class α_i. But γ will also have cosine similarity conducted between all comments and γ itself, considering the possibility for a comment to contain a sufficient number of subqueries to maintain a high similarity between the full query and itself. The similarities found for each class construct a ranking system for the most similar classes that fit the query. Figure 3 shows the methodology of query segmentation and ranking retrieval for RecBERT.

For each class which contains a set of comments, let its original query channel similarity S_1 be defined as the maximum similarity found with KNN search for the aggregate query. Let knnbe defined as the operator for KNN search.

$$S_1 = \max(\cos\_\text{sim}(e(\gamma), e(A)))\,; \forall A \subseteq \text{knn}(e(\gamma)) \tag{3}$$

For each class which contains a set of comments, let its raw subquery channel similarity s_2 be defined as the average of the maximum similarity found with KNN search for each subquery.

$$s_2 = \frac{1}{n}\sum_{i=1}^{n} \max(\cos\_\text{sim}(e(Q_i), e(B)))\,; \forall B \subseteq \text{knn}(e(Q_i)) \tag{4}$$

s_2 is passed through $\tanh^{-1}(x)$, a concave up function on the interval $(0,1)$ to adjust for the fact that there must be a large proportion of subqueries matched by different comments in the same class to constitute the query being similar to the class. The adjusted similarity S_2 is clamped to a maximum of 1.

$$f(x) = \tanh^{-1}(x)\,; x \in (0,1) \tag{5}$$

$$S_2 = \max(1, f(s_2)) \tag{6}$$

The final similarity S is then defined as the maximum similarity between S_1 and S_2. Top ranked classes can then be found by sorting based on S.

$$S = \max(S_1, S_2) \tag{7}$$

4 Experimental result

To prepare the data for domain adapting BERT, a corpus of 112 000 total user reviews of varying length for 1000 different titles (classes) was collected from myanimelist. The dataset contained user comments labeled with the title it corresponded to. A 20 training-validation split was used, leaving 89 600 reviews for the training set, and 22 400 reviews for the testing set. Default truncation was performed to allow all reviews to have 128 tokens. Reviews were preprocessed by conducting wordpiece tokenization on the text, adding the CLS and SEP tokens, and encoding these tokens into indexes corresponding to the BERT vocabulary,

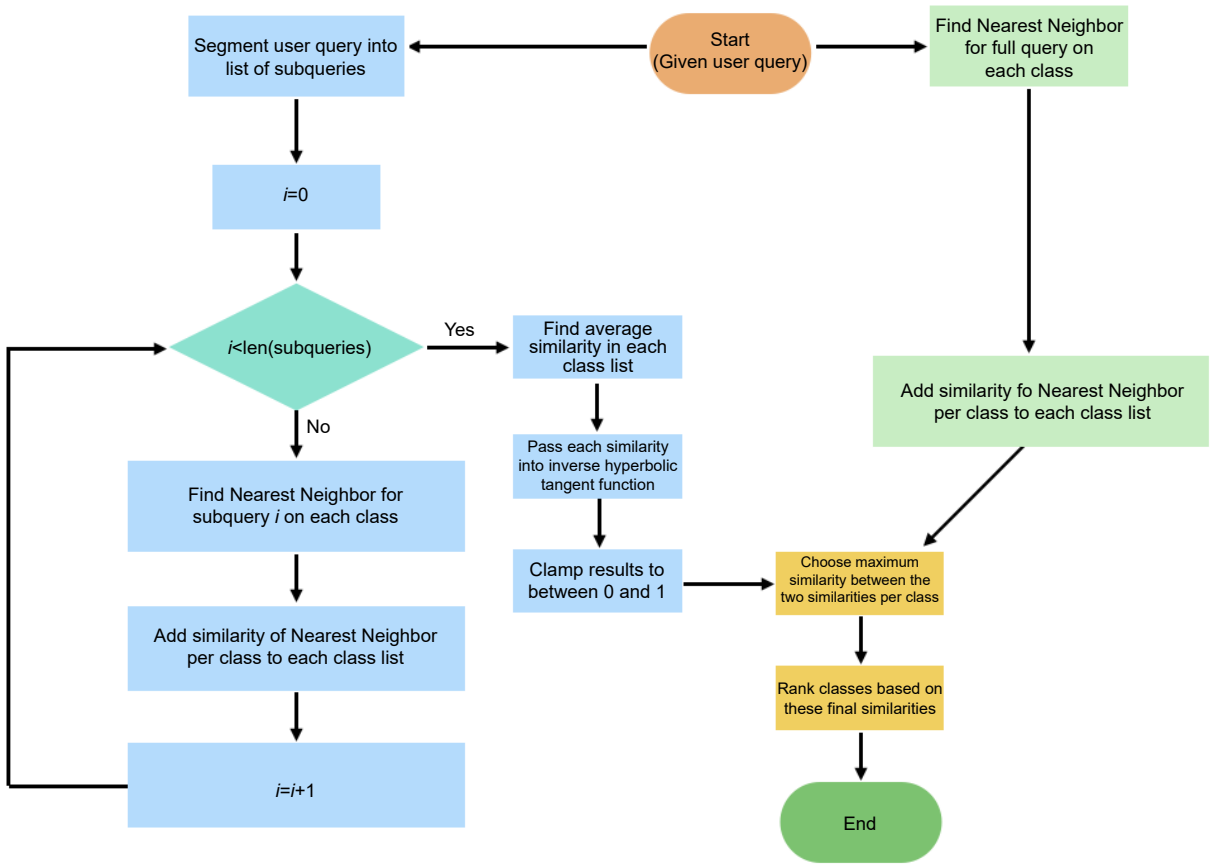


Fig. 3 Flowchart of query segmentation and ranking retrieval methodology.

Further pre-training was conducted on a distilBERT uncased model and RoBERTa[27] model respectively. Using a P100 GPU, further pre-training with Masked Language Modeling was conducted by having the model predict masked tokens in the user comments. A 15% masking probability was used, the same as proposed in the original BERT paper. Training was conducted for 10 epochs, using Adam optimizer with a 2×10^{-5} learning rate.

The RoBERTa model exhibited more accurate MLM results, approaching a loss of 1.55, while the distilBERT approached a loss of 1.88.

Next, the domain adapted transformer was fine tuned for document level similarity using SimCSE with MNR Loss, using the same user comments as when domain adapting the transformer base models. A batch size of 128 was used.

Tables 1 and 2 show the results of the recommendation system on comment class recognition. Tests are conducted on BERT, distilBERT[28], RoBERTa, distilRoBERTa, and RecBERT. RecBERT outperforms all state-of-the-art models in both the four-class classification and eight-class classification tasks. RecBERT outperforms the most precise state-of-the-art model (BERT) in precision by 1.63% when predicting between 4 classes, and outperforms the most precise state-of-the-art model (distilRoBERTa) in precision by 6.97% when predicting between 8 classes. As shown by these results, RecBERT's effectiveness increases when predicting between larger amounts of classes. This may be attributed to the need for more precise context understanding as classification is conducted between more classes. The fine-tuning on RecBERT with these user comments allows the model to understand the necessary context for the words in the specified domain. Figure 4 shows the confusion matrix and multi-class Receiver operating characteristic (ROC) curves for RecBERT in the four-class and eight-class classification tasks.

The t-distributed Stochastic Neighbor Embedding (t-SNE) visualization of user reviews in two-dimensional space[29] as shown in Fig. 5 demonstrates moderate

Table 1 Test results in comment class recognition for four classes. (%)

Model	Accuracy	Precision	Recall	F1 score
BERT cased	78.33	78.19	78.09	78.13
distilBERT cased	77.22	77.33	76.99	76.98
RoBERTa (Baseline)	73.89	72.96	73.47	72.56
distilRoBERTa	75.00	73.54	74.51	73.61
RecBERT	**79.72**	**79.82**	**79.63**	**79.62**

Table 2 Test results in comment class recognition for eight classes. (%)

Model	Accuracy	Precision	Recall	F1 score
BERT cased	50.72	48.73	49.49	48.51
distilBERT cased	54.74	51.81	52.88	52.14
RoBERTa (Baseline)	54.02	52.66	51.71	49.75
distilRoBERTa	54.31	53.63	52.35	51.92
RecBERT	**61.21**	**60.60**	**60.58**	**60.12**

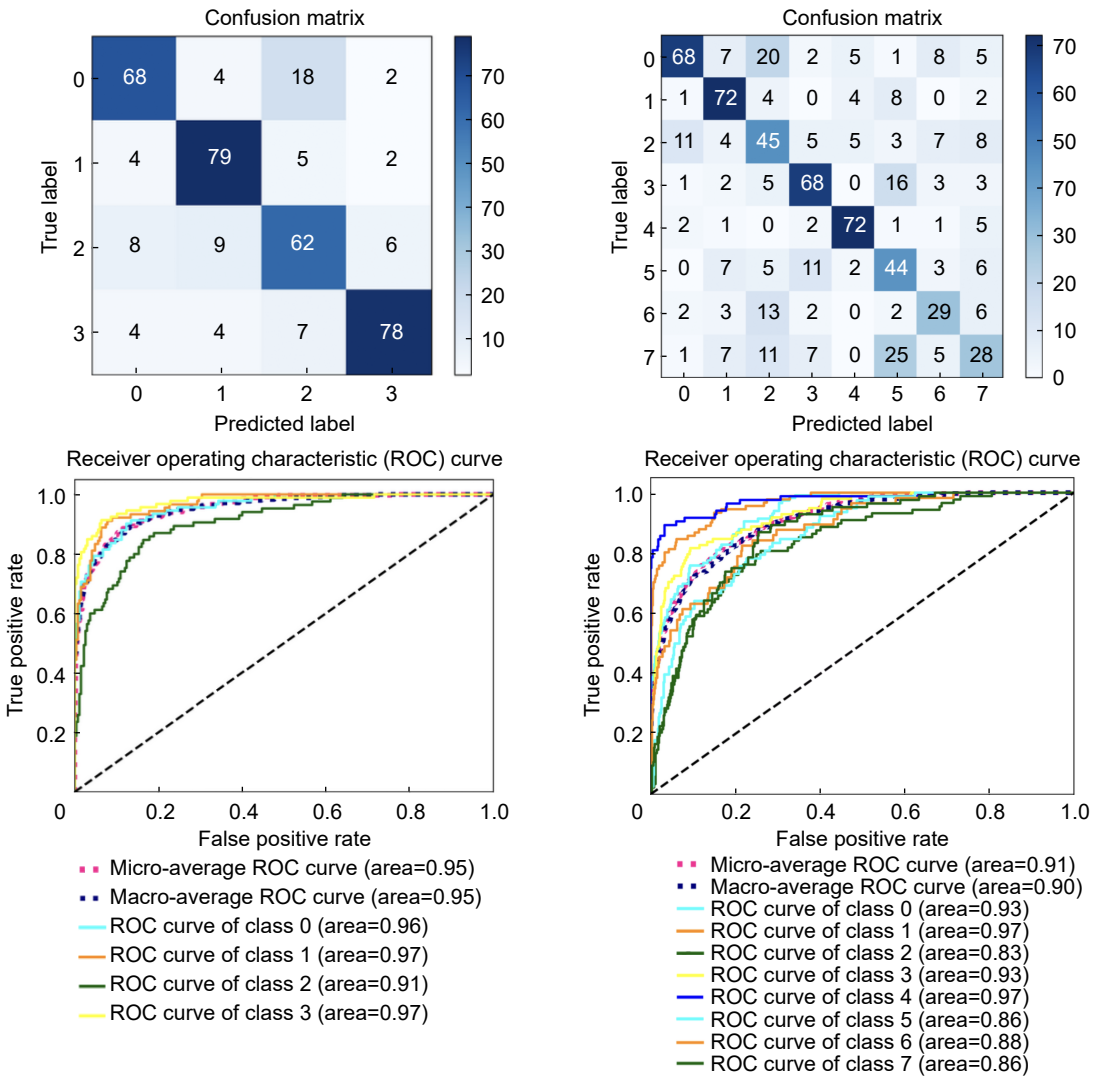


Fig. 4 Confusion matrix and multi-class ROC curves for RecBERT on four and eight classes, respectively.

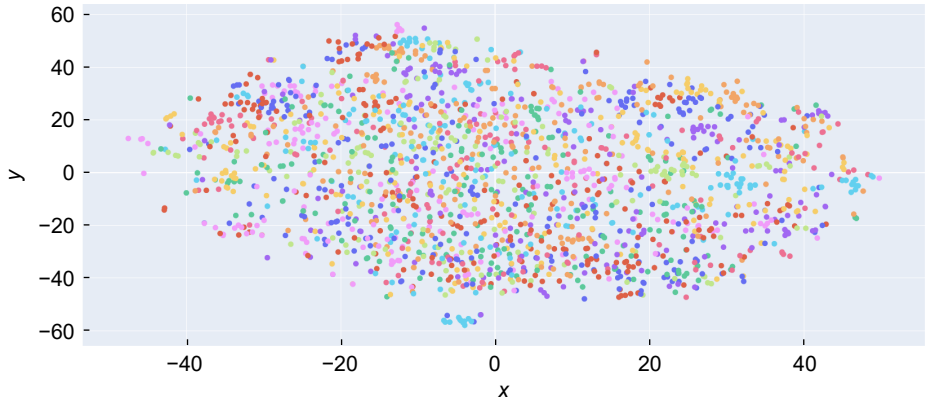


Fig. 5 t-SNE embeddings for 2000 comments.

clustering of embeddings based on classes. This is expected, as the comments belonging to a single class should not actually fully cluster int the first place, considering how different user comments should be providing different content to help satisfy the different potential subqueries within the query.

5 Conclusion

In this paper, RecBERT is introduced as a recommendation system composed of domain adaptation, fine-tuning of a sentence transformer, and query segmentation to enhance ranking retrieval. The usage of domain adaptation of BERT along with SimCSE with MNR to develop accurate sentence-level embeddings for comments is studied. This paper also explores the usage of LLM-enhanced query segmentation on the aggregate user query to construct a separate subquery similarity channel by performing separate KNN searches for each subquery and passing the average subquery similarity per class through the a concave up function, specifically the inverse hyperbolic tangent, to find an adjusted similarity, accounting for the fact that a larger number of queries being satisfied should increase the similarity in a non-linear fashion. The positive experimental results shown demonstrate the feasibility of this approach, as RecBERT achieved state-of-the-art accuracy, precision, recall, and F1 score on the myanimelist dataset for four and eight class classification respectively.

However, RecBERT's scalability remains a concern. KNN search becomes more computationally expensive as the number of comments to index increases in large-scale applications. In the future, there will be a focus on improving the scalability of RecBERT by using more approximate search algorithms. The accuracy of RecBERT may also be further improved by building a gold-standard dataset for domain adaptation, as some user comments may not be representative of the majority of comments, preventing convergence. Additionally, the sentence lengths of user comments may be standardized to prevent unaccounted variability from affecting similarity search results.

References

[1] Z. S. Harris, Distributional structure, in *Papers on Syntax*, H. Hiż Ed. Dordrecht, the Netherlands: Springer, 1981, pp. 3–22.

[2] S. W. Kim and J. M. Gil, Research paper classification systems based on TF-IDF and LDA schemes, *Hum. Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 30, 2019.

[3] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, in *Proc. 8th Int. Conf. Database Theory*, London, UK, 2001, pp. 420–434.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv: 1706.03762, 2017.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805, 2018.

[6] L. Minto, M. Haller, B. Livshits, and H. Haddadi, Stronger privacy for federated collaborative filtering with implicit feedback, in *Proc. 15th ACM Conf. Recommender Systems*, Amsterdam, the Netherlands, 2021, pp. 342–350.

[7] R. Chen, Q. Hua, Y. S. Chang, B. Wang, L. Zhang, and X. Kong, A survey of collaborative filtering-based

recommender systems: From traditional methods to hybrid methods based on social networks, *IEEE Access*, vol. 6, pp. 64301–64320, 2018.

[8] Y. Shi, M. Larson, and A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges, *ACM Comput. Surv.*, vol. 47, no. 1, p. 3, 2014.

[9] B. K. Mylavarapu, Collaborative filtering and artificial neural network based recommendation system for advanced applications, *J. Comput. Commun*, vol. 6, no. 12, pp. 1–14, 2018.

[10] G. Linden, B. Smith, and J. York, Amazon. com recommendations: Item-to-item collaborative filtering, *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[11] A. van den Oord, S. Dieleman, and B. Schrauwen, Deep content-based music recommendation, in *Proc. 26th Int. Conf. Neural Information Processing Systems - Volume 2*, Lake Tahoe, Nevada, 2013, pp. 2643–2651.

[12] J. Lian, F. Zhang, X. Xie, and G. Sun, CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems, in *Proc. 26th Int. Conf. World Wide Web Companion*, Perth, Australia, 2017, pp. 817–818.

[13] W. Zhao, B. Wang, M. Yang, J. Ye, Z. Zhao, X. Chen, and Y. Shen, Leveraging long and short-term information in content-aware movie recommendation via adversarial training, *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4680–4693, 2020.

[14] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[15] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction, *NPJ Digit. Med.*, vol. 4, no. 1, p. 86, 2021.

[16] H. Choi, J. Kim, S. Joe and Y. Gwon, Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks, in *Proc. 2020 25th Int. Conf. Pattern Recognition* (*ICPR*), Milan, Italy, 2021, pp. 5482–5487.

[17] T. Jiang, S. Huang, Z. Q. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, L. Zhang, and Q. Zhang, PromptBERT: Improving BERT sentence embeddings with prompts, in *Proc. 2022 Conf. Empirical Methods in Natural Language Processing*, Abu Dhabi, UAE, 2022, pp. 8826–8837.

[18] N. Reimers and I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing*, Hong Kong, China, 2019, pp. 3982–3992.

[19] T. Gao, X. Yao, and D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, arXiv preprint arXiv: 2104.08821, 2021.

[20] P. Röttger and J. Pierrehumbert, Temporal adaptation of BERT and performance on downstream document classification: Insights from social media, in *Findings of the Association for Computational Linguistics*: *EMNLP 2021*, M. F. Moens, X. Huang, L. Specia, and S. W. T. Yih, Eds. Kerrville, TX, USA: Association for Computational Linguistics, 2020, pp. 2400–2412.

[21] C. Liu, W. Zhu, X. Zhang, and Q. Zhai, Sentence part-enhanced BERT with respect to downstream tasks, *Complex Intell. Syst.*, vol. 9, no. 1, pp. 463–474, 2023.

[22] N. Kassner and H. Schütze, BERT-kNN: Adding a kNN search component to pretrained language models for better QA, in *Findings of the Association for Computational Linguistics*: *EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Kerrville, TX, USA: Association for Computational Linguistics, 2020, pp. 3424–3430.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. , Language models are few-shot learners, arXiv preprint arXiv: 2005.14165, 2020.

[24] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. , Constitutional AI: Harmlessness from AI feedback, arXiv preprint arXiv: 2212.08073, 2022.

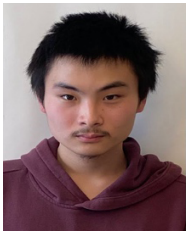
[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., LLaMA: Open and efficient foundation language models, arXiv preprint arXiv: 2302.13971, 2023.

[26] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, arXiv preprint arXiv: 2305.14314, 2023.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv: 1907.11692, 2019.

[28] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, arXiv preprint arXiv: 1910.01108, 2019.

[29] S. Arora, W. Hu, and P. K. Kothari, An analysis of the t-SNE algorithm for data visualization, in *Proc. 31st Conf. Learning Theory*, Stockholm, Sweden, 2018, pp. 1455–1462.



Richard Wu is with the Dublin Unified School District and President of the SF Artificial Intelligence Club, Dublin, CA, USA. His current research interests include natural language processing for semantic search and studying cryptography for supply chain management.