

QuOTeS: Query-Oriented Technical Summarization

Juan Ramirez-Orta¹ *, Eduardo Xamena^{2,3}, Ana Maguitman^{5,6}, Axel J. Soto^{5,6},
Flavia P. Zanoto⁴, and Evangelos Milios¹

¹ Department of Computer Science, Dalhousie University

² Institute of Research in Social Sciences and Humanities (ICSOH)

³ Universidad Nacional de Salta - CONICET

⁴ Escrever Ciência

⁵ Institute for Computer Science and Engineering, UNS - CONICET

⁶ Department of Computer Science and Engineering, Universidad Nacional del Sur

Abstract. When writing an academic paper, researchers often spend considerable time reviewing and summarizing papers to extract relevant citations and data to compose the *Introduction* and *Related Work* sections. To address this problem, we propose *QuOTeS*, an interactive system designed to retrieve sentences related to a summary of the research from a collection of potential references and hence assist in the composition of new papers. *QuOTeS* integrates techniques from Query-Focused Extractive Summarization and High-Recall Information Retrieval to provide Interactive Query-Focused Summarization of scientific documents. To measure the performance of our system, we carried out a comprehensive user study where participants uploaded papers related to their research and evaluated the system in terms of its usability and the quality of the summaries it produces. The results show that *QuOTeS* provides a positive user experience and consistently provides query-focused summaries that are relevant, concise, and complete. We share the code of our system and the novel Query-Focused Summarization dataset collected during our experiments at <https://github.com/jarobyte91/quotes>.

1 Introduction

When writing an academic paper, researchers often spend substantial time reviewing and summarizing papers to shape the *Introduction* and *Related Work* sections of their upcoming research. Given the ever-increasing number of academic publications available every year, this task has become very difficult and time-consuming, even for experienced researchers. A solution to this problem is to use Automatic Summarization systems, which take a long document or a collection of documents as input and produce a shorter text that conveys the same information.

The summaries produced by such systems are evaluated by measuring their fluency, coherence, conciseness, and completeness. To this end, Automatic Summarization systems can be divided into two categories, depending on their output.

* Please send correspondence to juan.ramirez.orta@dal.ca

In Extractive Summarization, the purpose of the system is to highlight or extract passages present in the original text, so the summaries are usually more coherent and complete. On the other hand, in Abstractive Summarization, the system generates the summary by introducing words that are not necessarily in the original text. Hence, the summaries are usually more fluent and concise. Although there have been significant advances recently [1], these complementary approaches share the same weakness: it is very hard for users to evaluate the quality of an automatic summary because it means that they have to go back to the original documents and verify that the system extracted the correct information.

Since evaluating summarization systems by hand is very difficult, several automatic metrics have been created with this purpose: BLEU [2], ROUGE [3], and METEOR [4] all aim to measure the quality of the summary produced by the system by comparing it with a reference summary via the distribution of its word n-grams. Despite being very convenient and popular, all these automatic metrics have a significant drawback: since they only look at the differences in the distribution of words between the system’s summary and the reference summary, they are not useful when the two summaries are worded differently, which is not necessarily a sign that the system is performing poorly.

Therefore, although Automatic Summarization systems display high performance when evaluated on benchmark datasets [5], they often cannot satisfy their users’ needs, given the inherent difficulty and ambiguity of the task [6]. An alternative approach to make systems more user-centric is Query-Focused Summarization [6], in which the users submit a query into the system to guide the summarization process and tailor it to their needs. Another alternative approach to this end is Interactive Summarization [7], in which the system produces an iteratively improved summary. Both of these approaches, and several others, take into account that the *correct* summary given a document collection depends on both the users and what they are looking for.

In this paper, we introduce *QuOTeS*, an interactive system designed to retrieve sentences relevant to a paragraph from a collection of academic articles to assist in the composition of new papers. *QuOTeS* integrates techniques from Query-Focused Extractive Summarization [6] and High-Recall Information Retrieval [8] to provide Interactive Query-Focused Summarization of scientific documents. An overview of how *QuOTeS* works and its components is shown in Fig. 1.

The main difficulty when creating a system like *QuOTeS* in a supervised manner is the lack of training data: gathering enough training examples would require having expert scientists carefully read several academic papers and manually label each one of their sentences concerning their relevance to the query, which would take substantial human effort. Therefore, we propose *QuOTeS* as a self-service tool: the users supply their academic papers (usually as PDFs), and *QuOTeS* provides an end-to-end service to aid them in the retrieval process. This paper includes the following contributions:

- A novel Interactive Query-Focused Summarization system that receives a short paragraph (called query) and a collection of academic documents as input and returns the sentences related to the query from the documents

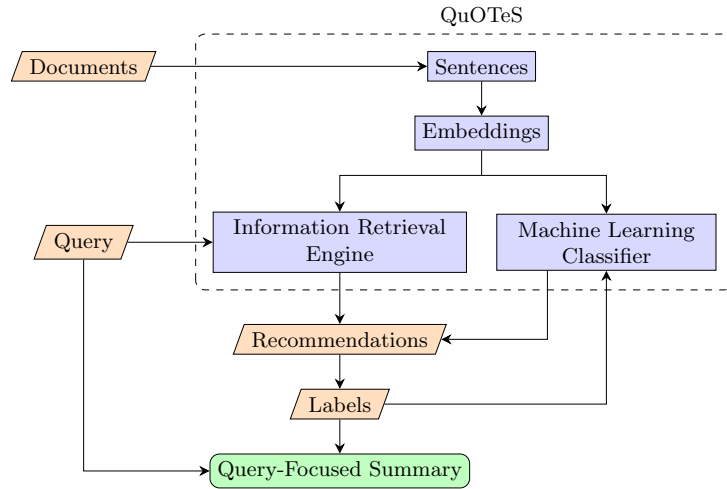


Fig. 1: Overview of how *QuOTeS* works. First, the user inputs their documents into the system, which then extracts the text present in them. Next, the system splits the text into sentences and computes an embedding for each one of them. After that, the user inputs their query, which is a short paragraph describing their research, and the system retrieves the most relevant sentences using the traditional *Vector Space Model*. The user then labels the recommendations and trains the system using techniques from High-Recall Information Retrieval to retrieve more relevant sentences until he or she is satisfied. Finally, the sentences labeled as relevant are returned to the user as the Query-Focused Summary of the collection.

in the collection. The system extracts the text directly from the academic documents provided by the user at runtime, minimizing the effort needed to perform complex queries on the text present in the documents. Finally, the system features techniques from High-Recall Information Retrieval to maximize the number of relevant sentences retrieved.

- A novel dataset composed of $(Query, Document\ Collection)$ pairs for the task of Query-Focused Summarization of Scientific Documents, each one with five documents and hundreds of sentences, along with the relevance labels produced by real users.
- A comprehensive analysis of the data collected during a user study of the system, where the system was evaluated using the System Usability Scale [9] and custom questionnaires to measure its usability and the quality of the summaries it produces.

2 Related Work

2.1 Query-Focused Summarization

The task of Query-Focused Summarization (QFS) was introduced in the 2005 Document Understanding Conference (DUC 2005) [6]. The focus of the conference was to develop new evaluation methods that take into account the variation of summaries produced by humans. Therefore, DUC 2005 had a single, user-oriented, question-focused summarization task that allowed the community to put some time and effort into helping with the new evaluation framework. The summarization task was to synthesize a well-organized and fluent answer to a complex question from a set of 25 to 50 documents. The relatively generous allowance of 250 words for each answer revealed how difficult it was for the systems to produce good multi-document summaries. The two subsequent editions of the conference (DUC 2006 [10] and DUC 2007 [11]) further enhanced the dataset produced in the first conference and have become the reference benchmark in the field.

Surprisingly, state-of-the-art algorithms designed for QFS do not significantly improve upon generic summarization methods when evaluated on traditional QFS datasets, as was shown in [12]. The authors hypothesized that this lack of success stems from the nature of the datasets, so they defined a novel method to quantify their Topic Concentration. Using their method, which is based on the ratio of sentences within the dataset that are already related to the query, they observed that the DUC datasets suffer from very high Topic Concentration. Therefore, they introduced TD-QFS, a new QFS dataset with controlled levels of Topic Concentration, and compared competitive baseline algorithms on it, reporting a solid improvement in performance for algorithms that model query relevance instead of generic summarizers. Finally, they presented three novel QFS algorithms (RelSum, ThresholdSum, and TFIDF-KLSum) that outperform, by a large margin, state-of-the-art QFS algorithms on the TD-QFS dataset.

A novel, unsupervised query-focused summarization method based on random walks over the graph of sentences in a document was introduced in [13]. First, word importance scores for each target document are computed using a word-level random walk. Next, they use a siamese neural network to optimize localized sentence representations obtained as the weighted average of word embeddings, where the word importance scores determine the weights. Finally, they conducted a sentence-level query-biased random walk to select a sentence to be used as a summary. In their experiments, they constructed a small evaluation dataset for QFS of scientific documents and showed that their method achieves competitive performance compared to other embeddings.

2.2 High-Recall Information Retrieval

A novel evaluation toolkit that simulates a human reviewer in the loop was introduced in [8]. The work compared the effectiveness of three Machine Learning protocols for Technology-Assisted Review (TAR) used in document review for

legal proceedings. It also addressed a central question in the deployment of TAR: should the initial training documents be selected randomly, or should they be selected using one or more deterministic methods, such as Keyword Search? To answer this question, they measured Recall as a function of human review effort on eight tasks. Their results showed that the best strategy to minimize the human effort is to use keywords to select the initial documents in conjunction with deterministic methods to train the classifier.

Continuous Active Learning achieves high Recall for TAR, not only for an overall information need but also for various facets of that information, whether explicit or implicit, as shown in [14]. Through simulations using Cormack and Grossman’s Technology-Assisted Review Evaluation Toolkit [8], the authors showed that Continuous Active Learning, applied to a multi-faceted topic, efficiently achieves high Recall for each facet of the topic. Their results also showed that Continuous Active Learning may achieve high overall Recall without sacrificing identifiable categories of relevant information.

A scalable version of the Continuous Active Learning protocol (S-CAL) was introduced in [15]. This novel variation requires $O(\log(N))$ labeling effort and $O(N\log(N))$ computational effort — where N is the number of unlabeled training examples — to construct a classifier whose effectiveness for a given labeling cost compares favorably with previously reported methods. At the same time, S-CAL offers calibrated estimates of Class Prevalence, Recall, and Precision, facilitating both threshold setting and determination of the adequacy of the classifier.

2.3 Interactive Query-Focused Summarization

A novel system that provides summaries for Computer Science publications was introduced in [16]. Through a qualitative user study, the authors identified the most valuable scenarios for discovering, exploring, and understanding scientific documents. Based on these findings, they built a system that retrieves and summarizes scientific documents for a given information need, either in the form of a free-text query or by choosing categorized values such as scientific tasks, datasets, and more. The system processed 270,000 papers to train its summarization module, which aims to generate concise yet detailed summaries. Finally, they validated their approach with human experts.

A novel framework to incorporate users’ feedback using a social robotics platform was introduced in [17]. Using the *Nao* robot (a programmable humanoid robot) as the interacting agent, they captured the user’s expressions and eye movements and used it to train their system via Reinforcement Learning. The whole approach was then evaluated in terms of its adaptability and interactivity.

A novel approach that exploits the user’s opinion in two stages was introduced in [18]. First, the query is refined by user-selected keywords, key phrases, and sentences extracted from the document collection. Then, it expands the query using a Genetic Algorithm, which ranks the final set of sentences using Maximal Marginal Relevance. To assess the performance of the proposed system, 45 graduate students in the field of Artificial Intelligence filled out a questionnaire after using the system on papers retrieved from the Artificial Intelligence category

of The Web of Science. Finally, the quality of the final summaries was measured in terms of the user’s perspective and redundancy, obtaining favorable results.

3 Design Goals

As shown in the previous section, there is a clear research gap in the literature: on the one hand, there exist effective systems for QFS, but on the other hand, none of them includes the user’s feedback about the relevance of each sentence present in the summary. On top of that, the task of QFS of scientific documents remains a fairly unexplored discipline, given the difficulty of extracting the text present in academic documents and the human effort required to evaluate such systems, as shown by [13]. Considering these limitations and the guidelines obtained from an expert consultant in scientific writing from our team, we state the following design goals behind the development of *QuOTeS*:

1. **Receive a paragraph query and a collection of academic documents as input and return the sentences relevant to the query from the documents in the collection.** Unlike previous works, *QuOTeS* is designed as an assistant in the task of writing *Introduction* and *Related Work* sections of papers in the making. To this end, the query inputted into the system is a short paragraph describing the upcoming work, which is a much more complex query than the one used in previous systems.
2. **Include the user in the retrieval loop.** As shown by previous works, summarization systems benefit from being interactive. Since it is difficult to express all the information need in a single query, the system needs to have some form of adaptation to the user, either by requiring more information about the user’s need (by some form of query expansion) or by incorporating the relevance labeling in the retrieval process.
3. **Provide a full end-to-end user experience in the sentence extraction process.** So far, query-focused summarization systems have been mainly evaluated on data from the DUC conferences. A usable system should be able to extract the text from various documents provided by the user, which can only be determined at runtime. Since the main form to distribute academic documents is PDF files, the system needs to be well adapted to extract the text in the different layouts in academic publications.
4. **Maximize Recall in the retrieval process.** Since the purpose of the system is to help the user retrieve the (possibly very) few relevant sentences from the hundreds of sentences in the collection, Recall is the most critical metric when using a system like *QuOTeS*, as users can always refine the output summary to adapt it to their needs. Therefore, we use Continuous Active Learning [8] as the training procedure for the classifier inside *QuOTeS*.

4 System Design

QuOTeS is a browser-based interactive system built with *Python*, mainly using the *Dash* package [19]. The methodology of the system is organized into seven

steps that allow the users to upload, search and explore their documents. An overview of how the steps relate to each other is shown in Fig. 2.

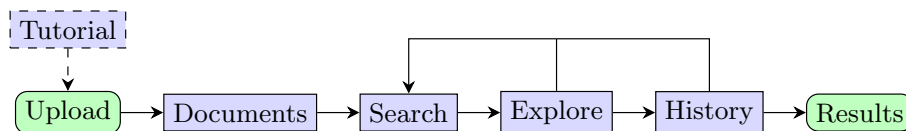


Fig. 2: Methodology of the system and its workflow.

4.1 Tutorial

In this step, the user can watch a 5-minute video¹ explaining the task that *QuOTeS* was made for and an overview of how to use the system. The main part of the video explains the different parts of the system and how they are linked together. It also explains the effect of the different retrieval options and how to download the results from the system to keep analyzing them. Since users will not necessarily need to watch the video every time they use the system, the first step they see when they access the website is the *Upload*, described below.

4.2 Upload

In this step, the users can upload their documents and get the system ready to start interacting with them via a file upload form. Once the text from all the documents has been extracted, they can click on *Process Documents* to prepare the system for the retrieval process. After that, they can select the options for the system in the *Settings* screen, which contains two drop-down menus. In the *Embeddings* menu, the user can choose how the system represents the query and the documents from three options: TFIDF embeddings based on word unigrams, TFIDF embeddings based on character trigrams and Sentence-BERT embeddings [20]. In the *Classifier* menu, the user can choose which Supervised Machine Learning algorithm to use as the backbone for the system from three options: Logistic Regression, Random Forest, and Support Vector Machine.

4.3 Documents

In this step, the user can browse the text extracted from the documents. The sentences from the papers are shown in the order they were found so that the user can verify that the text was extracted correctly. The user can select which documents to browse from the drop-down menu at the top, which displays all the documents that have been uploaded to the system. Later on, when the user starts labeling the sentences with respect to the query, they are colored accordingly: green (for relevant) or pink (for irrelevant).

¹ The video can be watched here: <https://www.youtube.com/watch?v=zR9XisDFQ7w>

4.4 *Search*

This is the first main step of the system. In the text box, users can write their query. After clicking on *Search*, the system retrieves the most relevant sentences using the classical *Vector Space Model* from Information Retrieval.

The sentences below are the best matches according to the query and the representation the user picked in the *Upload* step. The user can label them by clicking on them, which are colored accordingly: green (for relevant) or pink (for irrelevant). Once the users label the sentences, they can click on *Submit Labels*, after which the system records them and shows a new batch of recommendations.

4.5 *Explore*

This is the second main step of the system. Here, the system trains its classifier using the labels the user submits to improve its understanding of the query. Two plots at the top show the distribution of the recommendation score and how it breaks down by document to help the user better understand the collection. The sentences below work exactly like in *Search*, allowing the user to label them by clicking on them and submitting them into the system by clicking on *Submit Labels*. Users can label the collection as much as they want, but the recommended criterion is to stop when the system has not recommended anything relevant in three consecutive turns, shown in the colored box at the top right.

4.6 *History*

In this step, users can review what they have labeled and where to find it in the papers. The sentences are shown in the order they were presented to the user, along with the document they came from and their sentence number to make it easier to find them. Like before, the user can click on a sentence to relabel it if necessary, which makes it change color accordingly. There are two buttons at the top: *Clear* allows the user to restart the labeling process, and *Download .csv* downloads the labeling history as a CSV file for further analysis.

4.7 *Results*

In the last step of *QuOTeS*, the user can assess the results. There are two plots at the top that show the label counts and how they break down by document, while the bottom part displays the query and the sentences labeled as relevant. The query along these sentences make up the final output of the system, which is the Query-Focused Summary of the collection. The user can download this summary as a *.txt* file or the whole state of the system as a JSON file for further analysis.

5 Evaluation

To evaluate the effectiveness of *QuOTeS*, we performed a user study where each participant uploaded up to five documents into the system and labeled the

sentences in them for a maximum of one hour. The user study was implemented as a website written using the *Flask* package [21], where the participants went through eight screens to obtain their consent, explain the task to them and fill out a questionnaire about their perception of the difficulty of the task and the performance of *QuOTeS*. An overview of the user study is shown in Figure 3.

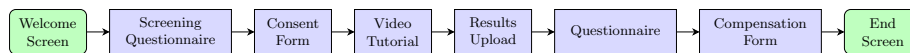


Fig. 3: Overview of the user study.

5.1 Methodology

In the *Welcome Screen*, the participants were shown a quick overview of the whole user study and its duration. In the *Screening Questionnaire*, they filled out a short questionnaire indicating their education level and the frequency they read academic papers. In the *Consent Form* screen, they read a copy of the consent form and agreed to participate by clicking on a checkbox at the end. In the *Video Tutorial* screen, they watched a five-minute video about the task and how to use *QuOTeS*. In the *Results Upload* screen, they were redirected to the website of *QuOTeS* and after using the system for a maximum of one hour, they uploaded the JSON file containing the state of the system at the end of their interaction. In the *Questionnaire* screen, they filled in a three-part questionnaire to evaluate the usability of *QuOTeS*, its features and the quality of the summaries. In the *Compensation Form*, they provided their name and email to be able to receive the compensation for their participation. Finally, the *End Screen* indicated that the study was over and they could close their browser.

5.2 Participants

To recruit participants, we sent a general email call to our faculty, explaining the recruiting process and the compensation. To verify that participants were fit for our study, they filled out a screening questionnaire with only two questions, with the purpose of knowing their research experience and the frequency they normally read academic papers. The requirements to participate were to have completed at least an undergraduate degree in a university and to read academic papers at least once a month. The results of the screening questionnaire for the participants who completed the full study are shown in Table 1, while the full results of the screening questionnaire can be found in the code repository.

5.3 Research Instrument

During the user study, the participants filled out a questionnaire composed of thirty questions divided into three parts: *Usability*, *Features*, and *Summary*

Table 1: Responses of the Screening Questionnaire from the participants that completed the study.

Paper Reading Frequency	Education	
	Undergraduate	Graduate
Every day	1	4
At least once a week	2	3
At least once every two weeks	0	1
At least once a month	3	1

Quality. In the *Usability* part, they filled out the questionnaire from the standard *System Usability Scale* [9], which is a quick and simple way to obtain a rough measure of the perceived usability of the system in the context of the task it is being used for. In the *Features* part, they answered sixteen questions about how difficult the task was and the usefulness of the different components of the system. In the *Summary Quality* part, they answered four questions about the relevance of the sentences in the system and the conciseness, redundancy, and completeness of the summaries produced. Finally, the participants submitted their opinions about the system and the user study in a free-text field. The full questionnaire presented to the participants can be found in the code repository.

5.4 Experimental Results

The frequency tables of the responses for the *System Usability Scale* questionnaire, the *Features* questionnaire, and the Summary Quality questionnaire can be found in the code repository. To make it easier to understand the responses from the questionnaires, we computed a score for the Features and Summary Quality parts in the same fashion as for the System Usability Scale: the questions with positive wording have a value from 0 to 4, depending on their position on the scale. In contrast, the questions with negative wording have a value from 4 to 0, again depending on their position on the scale. The distribution of the scores obtained during the user study is shown in Fig. 4.

6 Discussion

6.1 Questionnaire Responses

Overall, *QuOTeS* received a positive response across users, as the questionnaires show that the system seems to fulfill its purpose. Most of the time, the participants reported that the sentences recommended by the system seemed relevant and that the summaries appeared succinct, concise, and complete. Participants felt they understood the system’s task and how it works. Furthermore, they felt that the components of the system were useful. Nonetheless, the system can be improved in the following ways:

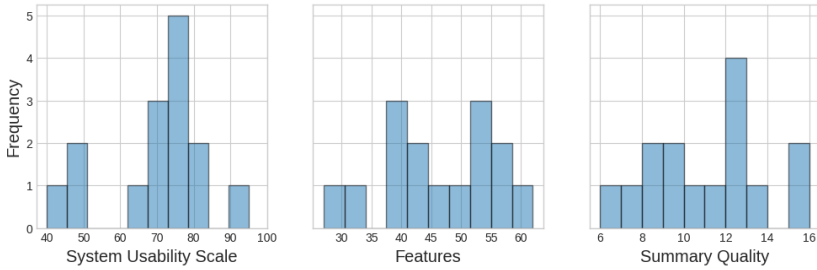


Fig. 4: Distribution of the questionnaire scores obtained during the user study. The possible range for each one of the scores is the following: *System Usability Scale* ranges from 0 to 100, with a mean of 69.67 and a median of 75; the *Features* score ranges from 0 to 64 with a mean of 45.87 and a median of 45; and the *Summary Quality* ranges from 0 to 16 with a mean of 10.67 and a median of 11. These results show that the users perceived the system as useful and well-designed and that the summaries it produces are adequate for the task.

- As shown by the last question of the *System Usability Scale* questionnaire, participants felt that they needed to learn many things before using the system. This is understandable, as *QuOTeS* is based on several concepts which are very specific to Natural Language Processing and Information Retrieval: the task of Query-Focused Summarization itself, the concept of embedding documents as points in space, and the concept of training a Machine Learning classifier on the fly to adapt it to the needs of the user. Nonetheless, knowledge of these concepts is not strictly required to obtain useful insights from the system.
- As shown by the *Features* questionnaire, the system can still be improved in terms of speed. Also, the users felt it was unclear what the different settings do and how to interpret the information in the plots. This may be improved with a better deployment and a better introductory tutorial that provides use cases for each one of the options in the settings: giving the user some guidance about when it is best to use word unigrams, character trigrams, and Sentence-BERT embeddings would facilitate picking the correct options.

The relationship between the different scores computed from the responses of the user study is shown in Fig. 5. All the scores show a clear, positive relationship with each other, with some outliers. The relationships found here are expected because all these scores are subjective and measure similar aspects of the system. Of all of them, the relationship between the System Usability Scale and the Summary Quality is the most interesting: it shows two subgroups, one in which the usability remains constant and the summary quality varies wildly, and another in which they both grow together. This may suggest that for some users, the query is so different from the collection that, although the system feels useful, they are dissatisfied with the results.

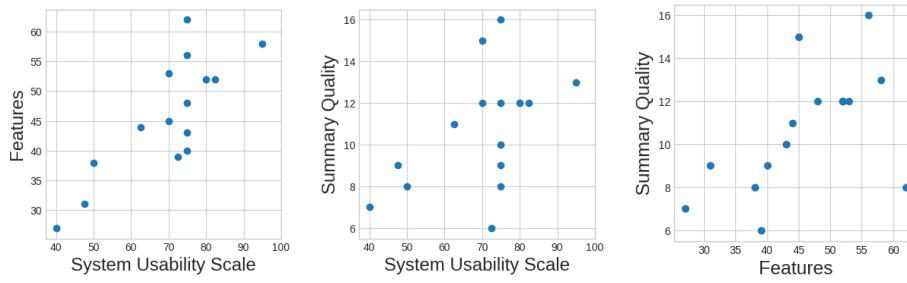


Fig. 5: Relationship between the scores computed from the questionnaires.

6.2 Analysis of the Labels Collected During the User Study

To further evaluate the performance of *QuOTeS*, we estimated the Precision and Topic Concentration using the data labeled by the users. To compute the Precision, we divided the number of sentences labeled as relevant over the total number of sentences shown to the user. To compute the Topic Concentration, we followed the approach from [12], using the Kullback-Leibler Divergence [22] between the unigram-based vocabulary of the document collection and the unigram-based vocabulary of the query-focused summary produced.

The distributions of the Precision and KL-Divergence, along with their relationship, are shown in Fig. 6. The relationship between the two metrics is noisy, but it is somewhat negative, suggesting that as the KL-Divergence decreases, the Precision increases. This result makes sense because the KL-Divergence measures how much the query deviates from the contents of the document collection.

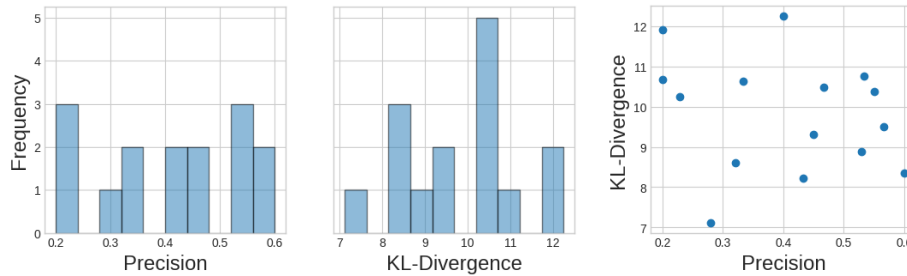


Fig. 6: Distributions of the Precision of the system (left) and the Kullback-Leibler Divergence between the word unigram distribution of the document collections and the summaries produced (center), along with their relationship (right).

On the other hand, Precision is displayed as a function of the Labeling Effort for each one of the participants in the user study in Fig. 7. We computed the Labeling Effort as the fraction of sentences reviewed by the user. The system

displays a stable average Precision of 0.39, which means that, on average, two out of five recommendations from the system are relevant. There appear to be two classes of users: in the first class, the system starts displaying a lot of relevant sentences, and the Precision drops as the system retrieves them; in the second class, the story is entirely the opposite: the system starts with very few correct recommendations, but it improves quickly as the user explores the collection.

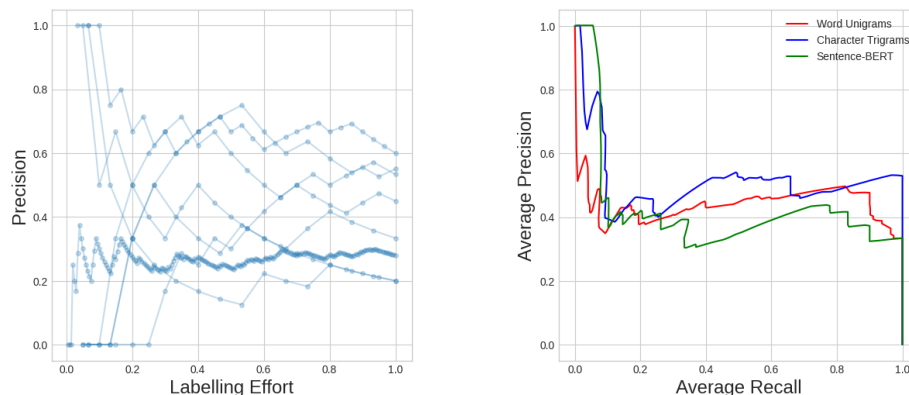


Fig. 7: Precision of the system. Precision as a function of the Labelling Effort for each one of the participants in the user study (left). Average Precision-Recall Curve of the different embeddings after removing the interactive component of *QuOTeS* (right).

The relationships between the Precision and the scores obtained from the questionnaires in the user study are shown in Fig. 8. Precision is well correlated with all the other scores, which is expected since it is the first metric perceived by the user, even before answering the questionnaires. An outlier is very interesting: one of the users gave the system low scores in terms of the questionnaires, despite having the highest Precision of the dataset. The labels produced by this user display a lower Divergence than usual, which means that his query was much closer to the collection than most users, as shown in Fig. 6. This could mean that he/she could already have excellent previous knowledge about the document collection. Therefore, although the system was retrieving relevant sentences, it was not giving the user any new knowledge.

The relationship between the Divergence and the scores is shown in Fig. 9. The relationship shown is noisier than the ones involving Precision. Although the System Usability Scale and Features scores show a positive relationship with the Divergence, this is not the case with the Summary Quality. This suggests that to have a high-quality summary, it is necessary to start with a collection close to the query. Another interesting point is that these relationships suggest that the system is perceived as more useful and better designed as the query deviates from the document collection.

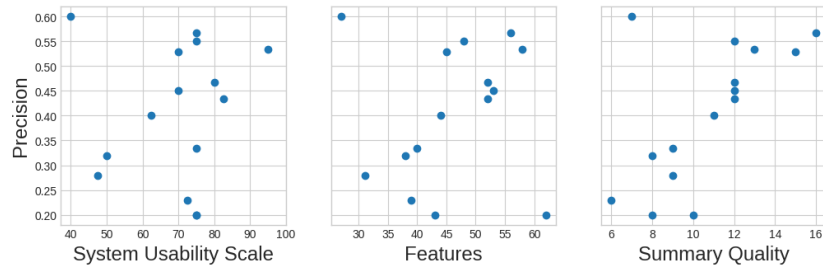


Fig. 8: Relation between the Precision of the system and the questionnaire scores.

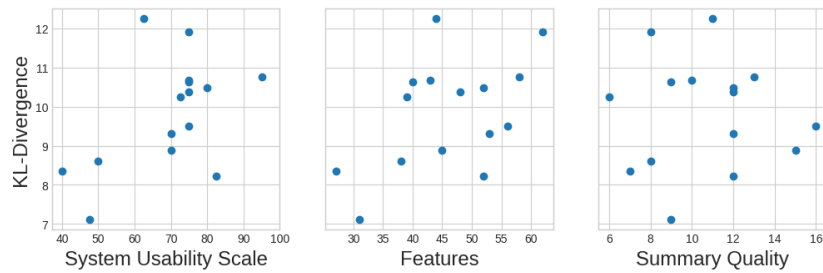


Fig. 9: Relationship between the Kullback-Leibler Divergence between the word unigram distribution of the document collection and produced summaries versus the questionnaire scores obtained in the user study.

To finalize our evaluation of *QuOTeS*, we measured its performance using the *(Query, Document Collection)* pairs collected during the user study. As a baseline, we used the traditional *Vector Space Model*, which is equivalent to disabling the *Machine Learning Classifier* component of *QuOTeS* (as shown in Fig. 1). We evaluated the three variations of the baseline system as they appear inside *QuOTeS*. The performance obtained by this baseline is shown in Fig. 7.

Even when using Sentence-BERT embeddings, the performance of the baseline system is markedly inferior compared to that of *QuOTeS*, as shown in Fig. 7. Although the Sentence-BERT embeddings start with a much higher Precision than the traditional embeddings, they quickly deteriorate as the score threshold increases, while the traditional embeddings catch up in terms of Precision with the same level of Recall. However, since none of these models obtained a satisfactory performance, it is clear that using *QuOTeS* enabled the users to find much more relevant sentences than they could have found otherwise. This highlights the importance of the Continuous Active Learning protocol in *QuOTeS*, as it enables the system to leverage the feedback from the user, so the results do not depend entirely on the embeddings produced by the language model.

6.3 Limitations

Although our experimental results are promising, the system we propose has two main limitations, given the complexity of the task and the amount of resources needed to produce benchmarks for this topic:

- First, the purpose of *QuOTeS* is not to provide fully automatic summaries since it is hard to guarantee that all the relevant sentences were retrieved in the process. Instead, its purpose is to point users in the right direction so that they can find the relevant information in the original documents.
- And second, the summaries produced by the system can still be improved using traditional techniques from Automatic Summarization. For example, their sentences in the summary could be reordered or removed to improve fluency and conciseness. These aspects would be beneficial if the goal is to produce a fully-automatic summary of the collection of articles.

7 Conclusions and Future Work

In this paper, we introduce *QuOTeS*, a system for Query-Focused Summarization of Scientific Documents designed to retrieve sentences relevant to a short paragraph, which takes the role of the query. *QuOTeS* is an interactive system based on the Continuous Active Learning protocol that incorporates the user’s feedback in the retrieval process to adapt itself to the user’s query.

After a comprehensive analysis of the questionnaires and labeled data obtained through a user study, we found that *QuOTeS* provides a positive user experience and fulfills its purpose. Also, the experimental results show that including both the user’s information need and feedback in the retrieval process leads to better results that cannot be obtained with the current non-interactive methods.

For future work, we would like to conduct a more comprehensive user study where users read the whole papers and label the sentences manually, after which they could use *QuOTeS* and compare the summaries produced. Another interesting future direction would be to compare the system heads-on with the main non-interactive methods from the literature on a large, standardized dataset.

8 Acknowledgements

We thank the Digital Research Alliance of Canada (<https://alliancecan.ca/en>), CIUNSa (Project B 2825), CONICET (PUE 22920160100056CO, PIBAA 2872021010 1236CO), MinCyT (PICT PRH-2017-0007), UNS (PGI 24/N051) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for the resources provided to enable this research.

References

1. Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: Pre-Training with Extracted Gap-Sentences for Abstractive Summarization. In

- Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org, 2020.
2. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
 3. Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
 4. Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
 5. Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
 6. Hoa Trang Dang. Overview of DUC 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.
 7. Anton Leuski, Chin-Yew Lin, and Eduard Hovy. iNeATS: Interactive multi-document summarization. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 125–128, Sapporo, Japan, July 2003. Association for Computational Linguistics.
 8. Gordon V. Cormack and Maura R. Grossman. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14*, page 153–162, New York, NY, USA, 2014. Association for Computing Machinery.
 9. John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
 10. Hoa Trang Dang. Overview of DUC 2006. In *Proceedings of the document understanding conference*, volume 2006, pages 1–10, 2006.
 11. Hoa Trang Dang. Overview of DUC 2007. In *Proceedings of the document understanding conference*, volume 2007, pages 1–53, 2007.
 12. Tal Baumel, Raphael Cohen, and Michael Elhadad. Topic Concentration in Query Focused Summarization Datasets. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, volume 30, 2016.
 13. Kazutoshi Shinoda and Akiko Aizawa. Query-focused scientific paper summarization with localized sentence representation. In *BIRNDL@ SIGIR*, 2018.
 14. Gordon V. Cormack and Maura R. Grossman. Multi-faceted recall of continuous active learning for technology-assisted review. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, page 763–766, New York, NY, USA, 2015. Association for Computing Machinery.
 15. Gordon V. Cormack and Maura R. Grossman. Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16*, page 1039–1048, New York, NY, USA, 2016. Association for Computing Machinery.

16. Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Francesca Bonin, and David Konopnicki. A summarization system for scientific documents. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 211–216, Hong Kong, China, November 2019. Association for Computational Linguistics.
17. Marzieh Zarinbal, Azadeh Mohebi, Hesamoddin Mosalli, Razieh Haratinik, Zahra Jabalameli, and Farnoush Bayatmakou. A new social robot for interactive query-based summarization: Scientific document summarization. In *Interactive Collaborative Robotics: 4th International Conference, ICR 2019, Istanbul, Turkey, August 20–25, 2019, Proceedings*, page 330–340, Berlin, Heidelberg, 2019. Springer-Verlag.
18. Farnoush Bayatmakou, Azadeh Mohebi, and Abbas Ahmadi. An interactive query-based approach for summarizing scientific documents. *Information Discovery and Delivery*, 50(2):176–191, 2021.
19. Plotly. Dash. Python package, <https://plotly.com/dash>, 2013. Visited on August 30, 2022.
20. Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
21. Pallets Projects. Flask: web development, one drop at a time. Python package, <https://flask.palletsprojects.com>, 2010. Visited on August 30, 2022.
22. Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.