LOCAL METHODS FOR DOCUMENT-LEVEL NATURAL LANGUAGE PROCESSING

by

JUAN ANTONIO RAMIREZ-ORTA

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
November 2023

To my parents and my family

Table of Contents

List of Tables

List of Figures

Abstract

Given the recent rise in popularity of methods based on Deep Neural Networks inside Natural Language Processing (NLP), important progress has been made in a variety of tasks that was not possible before, given their complexity and the heavy feature engineering required. However, the application of Deep Language Models to texts longer than a few paragraphs using standard hardware remains an open challenge.

In this thesis, we explore a novel set of techniques to process full documents that rely exclusively on local context. These techniques, called *local methods*, work by splitting the input into smaller pieces, processing them independently and combining the partial results in a coherent way. Their main advantage over other current methods is their efficiency: since they only process small parts of the full document, they prevent the model from wasting resources extracting meaningless relationships.

To test the effectiveness of local methods, we apply them in two tasks that require the processing full documents: the correction of documents processed with Optical Character Recognition systems and the Summarization of Scientific Documents. First, we introduce a method to summarize scientific documents of any length based on sentence embeddings and graphs that is simple, fast and efficient. Second, we introduce a method to correct long strings of characters by splitting them into n-grams, correcting them using character sequence-to-sequence models and joining them coherently via a voting. Third, we introduce a methodology for the Query-Focused Summarization of Scientific Documents based on splitting the input documents into sentences and training Machine Learning classifiers on-the-fly to determine their relevance to the query. And finally, we introduce a methodology to automatically obtain datasets for the tasks of Scientific Query-Focused Summarization and Citation Prediction by taking advantage of existing collections of academic documents.

In the end, the techniques introduced in this thesis provide evidence that local methods are a viable alternative to more complex, resource-hungry methods which currently represent the state of the art in NLP, promising to be resource- and sample-efficient, paving the way for a new family of methods for document-level NLP.

Acknowledgements

First, I would like to express my sincere thanks to my supervisor, Dr. Evangelos Milios, for all his support during my PhD, and for his flexibility during this whole time. I am very grateful for the opportunity to work with him and for all the things he has taught me over these years.

Also, my special thanks to my co-supervisors, Dr. Axel Soto and Dr. Ana Maguitman, for your support, patience and contributions to my work. Without you, completing my PhD would have been so much harder.

It is important for me to also thank a few other people: Jiawei, for all her love, support and for being there for me when I needed it the most; to my friends back in Mexico, Luis Javier, Kristofer, Renato, Karina and Axel, for reminding me about that little place called home; and to my friends in Canada, Sandra, Faezeh and Devi, for making this journey a lot more pleasant than I thought it would be.

Chapter 1

Introduction

1.1 Motivation

Given the recent introduction of methods based on Deep Neural Networks (DNNs) inside Natural Language Processing (NLP), significant progress has been made in various problems that was not possible before due to the heavy feature engineering required to effectively apply classical (not based in DNNs) Machine Learning (ML) methods.

Although the treatment of long sequences with Neural Networks has been an active area of research for decades, obtaining significant achievements like the LSTM [36], GRU [17] and Transformer [90] architectures, the processing of sequences with more than the well-established limit of 512 tokens like in BERT or GPT in domains with constrained hardware remains a difficult task [90, 28, 68].

Because of this, although Document Classification, Question Answering, and Summarization are some of the most interesting areas of Natural Language Processing (NLP), they are very challenging, as they involve processing full documents. Some of the reasons why processing full documents (also called document-level NLP) is difficult are the following: First, sometimes it is not clear how to split a full document into smaller pieces that are suitable for processing with standard methods. Second, the hardware requirements for document-level NLP are usually much higher than for other areas of NLP. And third, the collection of training examples for problems involving documents is even more expensive than for standard problems inside NLP, which becomes a major issue given that DNNs require large datasets to be trained.

More recently, the ML community has shifted their efforts to improve upon this obstacle, with benchmarks like the Long Range Arena [95] and novel architectures that improve upon the efficiency of the original Transformer, like the Sparse Transformer [76], the Longformer [37] and Big Bird [98]. These new architectures have provided evidence that in a long text, not all the token pairs are of interest in order to operate with the whole sequence, as their attention mechanisms follow different patterns to drop most connections and hence improve the efficiency of the model. Although effective, these models are still very resource-heavy, and require significant engineering to operate with documents longer than 4,096 tokens [37].

Unlike the current efficient Transformer architectures, this thesis tackles the treatment of full documents by introducing *local methods*. A method for document processing is called *local* if it operates with the whole sequence by splitting it into smaller parts, processing them independently and combining the partial results coherently. Their main advantage is that they allow to integrate into the algorithms a bias towards the range of the dependencies required to solve a task by only letting the model to see specific parts of the sequence. This bias encodes the human knowledge that, for some tasks, not all the information of the whole sequence is needed, and by taking this into account, the efficiency of the models can be dramatically improved while maintaining a good performance.

As a use case, we apply local methods to three areas of NLP which involve the processing of long sequences of text: the production of long extractive summaries from full scientific documents, the correction at the character level of documents processed with Optical Character Recognition (OCR) engines and the Query-Focused Summarization of Scientific Documents. Although it may seem that these areas require to process full documents as a single sequence, in this thesis we show the successful application of local methods in them. In the first case, we show that splitting the document into sentences and using methods from Graph Theory to find the most important sentences yields promising results. In the second case, we show that very long sequences of characters can be corrected using character sequence-to-sequence models that only correct small segments of the whole sequence. Finally, in the third case, we show that promising results can be obtained by splitting the documents into sentences and training classifiers to identify which sentences are relevant to the query.

In the end, the methods and results presented in this thesis provide evidence that local methods are a viable alternative to more complex, resource-hungry methods which currently represent the state of the art in NLP, promising to be resource- and sample-efficient, paving the way for a new family of methods for document-level NLP.

1.2 Contributions

- A novel unsupervised method for the extractive summarization of full scientific documents, based on the sentence embeddings produced by deep language models and techniques from Graph Theory. The method is simple, fast, can summarize documents of any length, and is able to produce summaries of any length, making it ideal for flexible, online implementations.

- A novel methodology to extend sequence-to-sequence models to process sequences much longer than the ones they were trained on. The methodology is based on splitting a long sequence into n-grams, processing them independently, and then combining them with a voting scheme that acts as a very large ensemble of sequence-to-sequence models, each one of them processing a different part of the sequence.

- A software suite for the correction of documents already processed with Optical Character Recognition (OCR) systems. The suite includes nine pre-trained models for Bulgarian, Czech, German, English, Spanish, French, Dutch, Polish and Slovak.

- A novel methodology for the Interactive Query-Focused Summarization of Scientific Documents. The methodology is embodied in an interactive system called *QuOTeS*, which receives a query and a collection of academic documents as input and outputs the query-focused summary of the collection, aided by the feedback provided by the user at run-time.

- A novel dataset for the Query-Focused Summarization of Scientific Documents composed of 8 examples, each one with query, a collection of full academic documents and the relevance labels for the sentences in the documents. The data was collected by real users during a comprehensive user study, where each of them uploaded documents relevant to their own research and manually labelled hundreds of sentences according to their relevance to the query.

- A novel methodology to automatically produce datasets for the tasks of Query-Focused Summarization of Scientific Documents and Citation Prediction using existing collections of academic documents. The methodology receives raw collections of academic papers and produces clean tables containing the papers in the collection, all the references found in these papers and the citations where each one of the references was mentioned. Together these three tables allow for the analysis of the whole collection and the creation of datasets for different NLP tasks.

- A novel dataset for the Query-Focused Summarization of Scientific Documents composed of 8,965 examples, each one with query, a collection of full academic documents and the relevance labels for the sentences in the documents. The dataset was automatically produced by applying the methodology previously mentioned to the papers from our reading group, composed originally of 1,091 academic papers.

1.3 Outline

The overall structure of this thesis is the following:

- **Chapter 2** describes a novel method to summarize full scientific papers in an unsupervised way by incorporating sentence embeddings produced by deep language models into graph-based techniques. The proposed technique has several advantages: it is simple, efficient, can summarize documents of any size and can produce summaries of any length. It also offers competitive performance when compared with more sophisticated supervised methods, and can be used as preparation step for more complex techniques.

- **Chapter 3** introduces a novel method to extend sequence-to-sequence models to process sequences much longer than the ones seen during training. As a use case, the method is applied on nine languages from the ICDAR 2019 competition on Post-OCR Text Correction, achieving a new state-of-the-art performance on five of them. After thorough experimentation, the strategy with the best performance involved splitting the input document into character n-grams and combining their individual corrections into the final output using a voting scheme that is equivalent to an ensemble of a large number of sequence models. The chapter closes with a thorough analysis of the results obtained in the experiments and an investigation on how to weigh the contributions from each one of the members of the ensemble.

- **Chapter 4** introduces a novel methodology for the Query-Focused Summarization of scientific papers. More specifically, the purpose of this methodology is to assist researchers in the composition of new papers, helping them to give shape to the *Introduction* and *Related Work* sections of their papers. The methodology is embodied in *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 effectiveness of the methodology, a comprehensive user study was carried out, 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 showed that *QuOTeS* provides a positive user experience and consistently provides query-focused summaries that are relevant, concise, and complete. The chapter closes by analyzing the relationship between the responses obtained from the questionnaires and the labeled data obtained from the users, and by pointing out the limitations and future directions to improve the methodology.

- **Chapter 5** introduces a novel methodology to automatically produce datasets for the tasks of Query-Focused Summarization of Scientific Papers and Citation Prediction (QFS/CP). The basic idea behind the methodology is to take advantage of existing collections of academic papers to obtain large-scale datasets for these tasks automatically. After applying it to the papers from our reading group, it introduces the first large-scale dataset for QFS/CP, composed of 8,695 examples, each composed of a query, the sentences of the full text from a paper and the relevance labels for each. An important result in this chapter is that, after testing several classical and state-of-the-art text representation models and classifiers on this data, it was found that these tasks are far from being solved, and that classical models outperformed modern pre-trained deep language models (sometimes by a large margin). The chapter closes by digging more deeply into why classical methods outperformed the current ones via an analysis of the labels in the dataset, and by comparing the results obtained in the data obtained automatically versus the data obtained by having users manually label queries along document collections.

- Finally, **Chapter 6** presents a summary of the findings from each chapter, as well as possible future research directions, which are beyond the scope of this thesis.

Chapter 2

Unsupervised Document Summarization using Pre-Trained Sentence Embeddings and Graph Centrality

This chapter describes our submission for the LongSumm task in SDP 2021 [1]. We propose a method for incorporating sentence embeddings produced by deep language models into extractive summarization techniques based on graph centrality in an unsupervised manner. The proposed method is simple, fast, can summarize any kind of document of any size and can satisfy any length constraints for the summaries produced. The method offers competitive performance to more sophisticated supervised methods and can serve as a proxy for abstractive summarization techniques. The code for the method introduced in chapter can be found at `https://github.com/jarobyte91/auto_summ`.

2.1 Introduction

Automatic text summarization is a very old and important task in Natural Language Processing (NLP) that has received continued attention since the creation of the field in the late 50's [34], mainly because of the ever-increasing size of document collections. The objective of the task is, given a document, to produce a shorter text with maximum information content, fluency and coherence. The summarization task can be classified into extractive and abstractive, where extractive summarization means that the summary is composed exclusively of passages present in the original document, while abstractive summarization means that there can be words in the summary that did not appear in the original document.

Since the creation of the first neural language models [11], vector representations of text that encode meaning (called embeddings) have played a significant role in NLP. They allow the application of statistical and geometrical methods to words,

[1] This chapter is an improved version of the paper [71], published in the LongSumm shared task of the 2nd Scholarly Document Workshop at NAACL 2021.

sentences and documents ([40], [58], [75]), leading to state-of-the-art performance on several NLP tasks like Information Retrieval, Question Answering or Paraphrase Identification. Among these neural language models, very deep pre-trained neural language models, like BERT [28], T5 [20], and GPT-3 [13] have shown impressive performance in tasks like language modelling and text generation or benchmarks like GLUE [92].

An important variation of extractive summarization that goes back as far as the late 90's [33, 32] utilizes graphs, where the nodes represent text units and the links represent some measure of semantic similarity. These early graph-based summarization techniques involved creating a graph where the nodes were the sentences or paragraphs of a document and two nodes were connected if the corresponding text units had a similar vocabulary. After creating the document graph, the system created a summary by starting at the first paragraph and following random walks defined by different algorithms that tried to cover as much of the graph as possible.

A more evolved approach was the creation of *lexical centrality* [30] [56] [93], which is a measure of the importance of a passage in a text where the sentences of the document are connected by the similarity of their vocabularies.

The current state of the art in automatic summarization with graphs is mainly based on algorithms like PageRank [84] enhanced with statistical information of the terms in the document (like in [69]) or Graph Neural Networks [45] on top of deep language models (like in [41]).

Only two systems from the previous Scholarly Document Processing Workshop (held in 2020) are based on graphs: CIST-BUPT and Monash-Summ.

In CIST-BUPT [52], they used Recurrent Neural Networks to create sentence embeddings that can be used to build a graph which is then fed into a Graph Convolutional Network [45] and a Graph Attention Network [91] to create extractive summaries. To generate abstractive summaries, they used the gap-sentence method of [42] to fine-tune T5 [20].

In Monash-Summ [43], they propose an unsupervised approach that leverages linguistic knowledge to construct a sentence graph like in SummPip [100]. The graph nodes, which represent sentences, are further clustered to control the summary length, while the final abstractive summary is created from the key phrases and discourse from each cluster.

Unlike previous methods, this work leverages the sentence embeddings produced by Pre-Trained Language Models and ideas from Graph Theory to produce full extractive summaries of any length from academic documents. The essential idea is that, while the sentence embeddings produced by SBERT [75] are not well suited for clustering algorithms like Hierarchical Clustering or DBSCAN [31], they produce excellent results for Paraphrase Identification or Semantic Textual Similarity when compared with Cosine Similarity, which implies that they can be used along with graph centrality methods. The text summarization method proposed in this paper has the following contributions:

- It is unsupervised and can be used as a proxy for more advanced summarization methods.
- Can easily scale to arbitrarily large amounts of text.
- Is fast and easy to implement.
- Can fit any length requirements for the production of summaries.

2.2 Methodology

In this section, we describe how the system works. The system is composed of three main steps: first, we use SBERT to produce sentence embeddings for every sentence in the document to summarize; next, we form a graph by comparing all the pairs of sentence embeddings obtained and finally, we rank the sentences by their degree centrality in this graph. Fig. 2.1 gives an overview of the whole method.

2.2.1 Sentence Tokenization

The first step of our pipeline is to split the input text into a list of sentences. This step is critical because if the sentences are too long, the final summary will have a lot of meaningless content (therefore losing precision). However, if the sentences are too short, there is a risk of not having enough context to produce an accurate

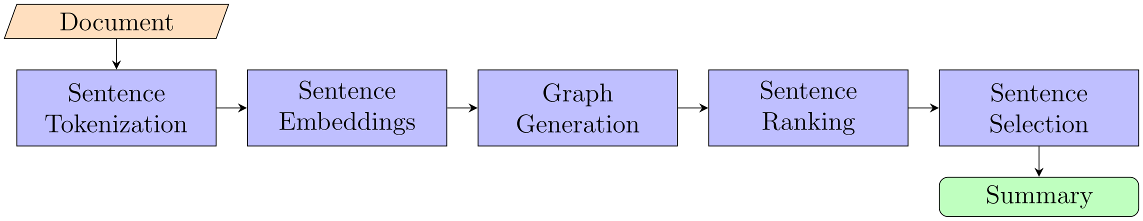

Figure 2.1: The pipeline of the proposed method. In the first step, we split the input text into sentences by using a regular expression handcrafted specifically for scientific documents. In the second step, we compute the sentence embeddings of the parsed sentences using SBERT. In the third step, we create a graph by comparing all the pairs of sentence embeddings obtained using cosine similarity. In the fourth step, we rank the sentences by the degree centrality in the generated graph. In the fifth and final step, we only keep a certain number of sentences or words to adjust to the length requirements of the summary.

sentence embedding for them or extracting meaningless sequences, like data in tables or numbers that lie in the middle of the text.

We found that the function `sent_tokenize()` from the NLTK package [87] often failed because of the numbers in the tables and the abbreviations, like *et al.*, which are very common in scientific literature. Because of this, we used a set of regular expressions handcrafted specifically to split the text found in scientific documents on top of the standard unsupervised tokenizer found in NLTK. The specific details of the implementation can be found here `https://github.com/jarobyte91/auto_summ/blob/master/engine/core/engine_summarization.py`.

2.2.2 Sentence Embeddings

After extracting the sentences, the next step is to produce the sentence embedding of each sentence using SBERT [75], which is a Transformer-based [90] model built on top of BERT [28] that takes as input sentences and produces sentence embeddings that can be compared with cosine similarity, which is given by the following formula:

$$sim(x, y) = \frac{x \cdot y}{|x||y|}.$$

As shown in [75], these sentence embeddings are superior in quality than taking the CLS token of BERT or averaging the sentence embeddings of the words in the sentence produced by BERT, GloVe [40], or Word2Vec [58].

SBERT, like BERT, was pre-trained on a general large text collection to learn good sentence embeddings, but it has to be fine-tuned on a more specific data set according to the task. Since we are working with scientific papers, we picked the *base* version of RoBERTa [96] that was fine-tuned in the MSMARCO data set [5] for the Information Retrieval task.

2.2.3 Graph Generation

After the sentence embeddings have been produced, the next step is to produce a weighted complete graph with a node for each sentence in the text. Its edges are weighted according to the cosine similarities of the corresponding sentence embeddings. An example graph is depicted in Fig. 2.2.

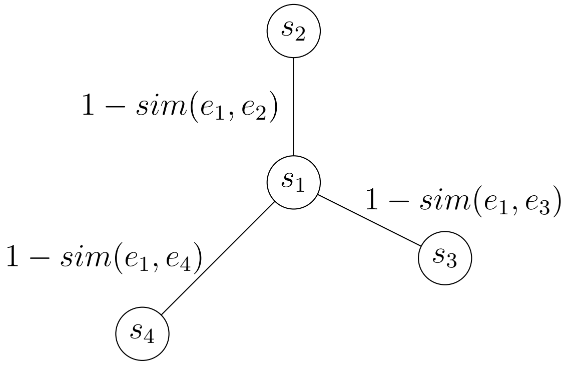

Figure 2.2: The process of graph generation and ranking of the sentences. Every node in the generated complete graph represents a sentence in the document and the weight of each edge is given by the similarity between the nodes it connects. The importance of the sentence in the document is modelled as $rank(s_i) = \sum_{j=1}^{n} 1 - sim(e_i, e_j)$, where e_i and e_j are the corresponding SBERT sentence embeddings of s_i and s_j.

To build this graph, the first step is to gather all the pairwise cosine similarities in a matrix. Let $D = (s_1, s_2, ..., s_n)$ be a document. Using SBERT, we produce a sequence of vectors $(e_1, e_2, ..., e_n)$, where e_i is the sentence embedding of s_i. Then, we can compute the matrix A, where $A[i, j] = 1 - sim(e_i, e_j)$.

We make the following observations:

- The diagonal of A is composed exclusively of zeros, because $A[i, i] = 1 - sim(e_i, e_i) = 0$.

- The matrix A is symmetric, because $A[i,j] = 1 - sim(e_i, e_j) = 1 - sim(e_j, e_i) = A[j,i]$.
- All the entries in A are non-negative, because $-1 \leq sim(e_i, e_j) \leq 1$.

These observations imply that the matrix A can be interpreted as the adjacency matrix of a weighted complete graph $G = (V, E)$ where $V = \{s_1, s_2, ..., s_n\}$, $E = \{(s_1, s_2)|s_1, s_2 \in V\}$ and the edges are weighted by the following function: $w(s_1, s_2) = 1 - sim(e_1, e_2)$.

2.2.4 Sentence Ranking

The forth step is to assign a score for each sentence that allows us to sort them by their importance in the document. As a consequence, we define the importance rank for each sentence as follows:

$$rank(s_i) = \sum_{j=1}^{n} A[i,j] = \sum_{j=1}^{n} 1 - sim(e_i, e_j), \tag{2.1}$$

where e_i and e_j are the corresponding SBERT sentence embedding for s_i and s_j.

To motivate this definition, we observe that adding the entries of the matrix A column-wise gives naturally a ranking of the nodes of G that is a natural generalization of the degree centrality. However, in our ranking, the most "central" sentences (sentences that are similar to many other sentences in the document) have lower scores than the ones that are less "central."

To further support this definition, we observe that if G were an undirected, unweighted simple graph $G = (V, E)$ (that is, the entries of A are either 0 or 1, A is symmetric and only has zeros in its diagonal), then we would have that

$$\sum_{j=1}^{n} A[i,j] = \#\{v \in V|(v_i, v) \in E\}, \tag{2.2}$$

which is the definition of the degree of node v_i and is clearly a (somewhat crude) measure of the importance of the node in the graph.

It is important to note that in scientific papers, which have around 300 sentences, the proposed method takes around 1 second for the whole process. This result implies that there is no obstacle for applying this method to longer documents since producing the sentence embeddings with the SBERT implementation is very efficient, and the only thing that we are doing is compare all the pairs of sentence embeddings, which can be done with highly efficient linear algebra libraries.

2.2.5 Sentence Selection

The final step in the method is to select the sentences that are going to form the summary. To do this, we can take only the bottom n-percentile in reverse (as opposed to the top n-percentile, since in our method, a lower rank means that the sentence is more important in the document) or concatenate the ranked sentences in reverse (so that the sentences with the lowest ranks -that is, the most important ones- come first) and take the first k words to satisfy a word-length constraint for the summaries.

2.3 Experimental Setup

2.3.1 Data

Since our method is for unsupervised extractive summarization, we only used the extractive summaries in the TalkSumm data set [51] to estimate the appropriate threshold value for the sentence selection phase. As suggested in the task, we used science-parse [3] to extract the text of the scientific articles and split it into sections. Given that the objective of the task is to produce long summaries for the documents, we discarded the title and abstract and then took as input for the algorithm the remaining text as a single block.

The dataset for this competition is composed of 1,705 papers along with their extractive summaries, from which 700 also include an abstractive summary. The dataset was built using a generative model trained on talks, presentations and blog posts about the papers in order to learn how to extract the most relevant content.

2.3.2 Evaluation

As is customary in summarization tasks, we used ROUGE [54] in its variations ROUGE-1, ROUGE-2 and ROUGE-L.

2.3.3 Percentile Threshold in the Selection Phase

We tried with $p = \{1, 1.5, 2, 2.5, 5, 10, 15\}$ as the value of the bottom percentage of sentences to keep for the final summary and truncated the output to satisfy the 600 word limit for the task when the summary was longer. It is important to note that the freedom of this parameter allows the system to produce summaries of arbitrary length, depending on the task at hand.

2.4 Results

Overall, we observed that the 600-word constraint of the task prevented our method from performing better, but we also observed that the best summaries produced by our method are too long (around 1,000 words or more). Table 2.1 displays the performance of the method variations that we submitted to the task.

Bottom %	ROUGE-1		ROUGE-2		ROUGE-L		Mean Length
	F-measure	Recall	F-measure	Recall	F-measure	Recall	
1.0	0.24	0.15	0.06	0.03	0.11	0.07	183.2
1.5	0.29	0.21	0.08	0.05	0.13	0.09	257.0
2.0	0.33	0.25	0.08	0.06	0.14	0.10	314.8
2.5	0.37	0.29	0.09	0.07	0.15	0.11	366.7
5.0	0.44	0.39	0.12	0.10	0.16	0.14	530.5
10.0	0.46	0.43	0.12	0.12	0.17	0.16	591.3
15.0	0.46	0.43	0.12	0.12	0.17	0.16	597.0

Table 2.1: Performance of the different variations of the proposed method submitted to the task. In this setting, the ranked sentences were sorted in reverse and concatenated to form a preliminary output, which was truncated at 600 words to comply with the task's requirements. The "Bottom %" column displays the percentile used in the sentence selection phase of the method. Mean length displays the average length in words of the summaries produced for the test set.

Table 2.2 displays the two best-performing models from the LongSumm competition. Both models are discussed with more detail in [10], but we give a short description below:

The N&E method is based on sessions, which are segments of the paper of a given size. Then, the method jointly trains an abstractive summarization model and a extractive summarization model to combine their output via an ensemble, and uses ROUGE [54] to make the model include sentences from the input to match the reference summary. The basic idea is that the sessions are a more flexible division than paragraphs or sections of the paper, and the abstractive and extractive models complement each other to improve the quality of the output.

The CNLP-NITS method is a variation of TextRank [57], which builds a graph where each node represents a sentence and the edges are weighted according to the content overlap between the corresponding sentences. To measure the content overlap between the sentences, they tested several similarity functions: Longest Common Substring, Cosine Similarity, BM25 [80] and BM25+. In their experiments, the best performing function was BM25. After that, they apply the well-known PageRank algorithm to model the importance of each sentence.

Team	ROUGE-1		ROUGE-2		ROUGE-L	
	F-Measure	Recall	F-Measure	Recall	F-Measure	Recall
N&E	0.5507	0.5660	0.1945	0.1998	0.2295	0.2357
CNLP-NITS	0.5131	0.5271	0.1610	0.1656	0.1916	0.1971
Dalhousie University	0.4621	0.4377	0.1280	0.1212	0.1701	0.1610

Table 2.2: The best-performing models from the LongSumm competition at the Second Scholarly Document Processing Workshop (SDP).

2.5 Conclusions and Future Work

The method introduced in this work displays competitive performance with more sophisticated methods and can be useful when there is not enough labelled data to train a deep neural summarization system while being fast, simple and efficient. Overall, we observed that the recall component of ROUGE for the proposed method has much room for improvement, as having sentences as the minimal text units makes it harder to include relevant phrases that are joined with others that are not so relevant. Another important future direction is to reduce the redundancy of the summaries, as it is common to have several versions of the same important sentence scattered across the document, so all these versions of the sentence appear in the final summary.

2.6 Appendix: Output Examples

1. **Sequence to Sequence Learning with Neural Networks.** We were able to do well on long sentences because we reversed the order of words in the source sentence but not the target sentences in the training and test set. So for example, instead of mapping the sentence a, b, c to the sentence α, β, γ, the LSTM is asked to map c, b, a to α, β, γ, where α, β, γ, is the translation of a, b, c. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Our work is closely related to Kalchbrenner and Blunsom [18], who were the first to map the input sentence into a vector and then back to a sentence, although they map sentences to vectors using convolutional neural networks, which lose the ordering of the words. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier. While the decoded translations of the LSTM ensemble do not outperform the best WMT'14 system, it is the first time that a pure neural translation system outperforms a phrase-based SMT baseline on a large scale MT. There are several variants of the BLEU score, and each variant is defined with a perl script. The LSTM is within 0.5 BLEU points of the best WMT'14 result if it is used to rescore the 1000-best list of the baseline system. 3.3 Reversing the Source Sentences While the LSTM is capable of solving problems with long term dependencies, we discovered that the LSTM learns much better when the source sentences are reversed (the target sentences are not reversed). When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The simplest strategy for general sequence learning is to map the input sequence to a fixed-sized vector using one RNN, and then to map the vector to the target sequence with another RNN (this approach has also been taken by Cho et al. The simple trick of reversing the words in the source sentence is one of the key technical contributions of

this work. By reversing the words in the source sentence, the average distance between corresponding words in the source and target language is unchanged. We were surprised by the extent of the improvement obtained by reversing the words in the source sentences. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our approach is closely related to Kalchbrenner and Blunsom [18] who were the first to map the entire input sentence to vector, and is related to Cho et al. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. They followed a similar approach, but they incorporated their NNLM into the decoder of an MT system and used the decoder's alignment information to provide the NNLM with the most useful words in the input sentence.

2. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.** For finetuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks. 3 Fine-tuning Procedure For fine-tuning, most model hyperparameters are the same as in pre-training, with the exception of the batch size, learning rate, and number of training epochs. To fine-tune on GLUE, we represent the input sequence (for single sentence or sentence pairs) as described in Section 3, and use the final hidden vector $C \in RH$ corresponding to the first input token ([CLS]) as the aggregate representation. To isolate the effect of these differences, we perform ablation experiments in Section 5.1 which demonstrate that the majority of the improvements are in fact coming from the two pre-training tasks and the bidirectionality they enable. 5.1 Effect of Pre-training Tasks We demonstrate the importance of the deep bidirectionality of BERT by evaluating two pretraining objectives using exactly the same pretraining data, fine-tuning scheme, and hyperparameters as BERTBASE : No NSP: A bidirectional model which is trained using the "masked LM" (MLM) but without the "next sentence prediction" (NSP) task. The core argument of this work is that the bi-directionality and the two pretraining tasks presented in Section 3.1 account for the majority of the empirical improvements, but we do note that

there are several other differences between how BERT and GPT were trained: • GPT is trained on the BooksCorpus (800M words); BERT is trained on the BooksCorpus (800M words) and Wikipedia (2,500M words). 5.3 Feature-based Approach with BERT All of the BERT results presented so far have used the fine-tuning approach, where a simple classification layer is added to the pre-trained model, and all parameters are jointly fine-tuned on a downstream task. The contextual representation of each token is the concatenation of the left-to-right and right-to-left representations. The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context. , 3072 for the H = 768 and 4096 for the H = 1024.4 We note that in the literature the bidirectional Trans2 Input/Output Representations To make BERT handle a variety of down-stream tasks, our input representation is able to unambiguously represent both a single sentence and a pair of sentences (e.g. For the feature-based approach, we concatenate the last 4 layers of BERT as the features, which was shown to be the best approach in Section 5.3. As shown in Figure 1, in the question answering task, we represent the input question and passage as a single packed sequence, with the question using the A embedding and the passage using the B embedding. For example, in OpenAI GPT, the authors use a left-toright architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer (Vaswani et al. 1 Illustration of the Pre-training Tasks We provide examples of the pre-training tasks in the following. Note that the purpose of the masking strategies is to reduce the mismatch between pre-training and fine-tuning, as the [MASK] symbol never appears during the fine-tuning stage. We use the representation of the first sub-token as the input to the token-level classifier over the NER label set. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). , 2017; Logeswaran and Lee, 2018), left-to-right generation of next sentence words given a representation of the previous sentence (Kiros et al. The best performing method concatenates the token representations from the top four hidden layers of the pre-trained Transformer, which is only 0.3 F1 behind fine-tuning the entire model.

3. **Improving Language Understanding by Generative Pre-Training.** By pre-training on a diverse corpus with long stretches of contiguous text our model acquires significant world knowledge and ability to process long-range dependencies which are then successfully transferred to solving discriminative tasks such as question answering, semantic similarity assessment, entailment determination, and text classification, improving the state of the art on 9 of the 12 datasets we study. A hypothesis is that the underlying generative model learns to perform many of the tasks we evaluate on in order to improve its language modeling capability and that the more structured 7 Table 5: Analysis of various model ablations on different tasks. For SST-2 (sentiment analysis), we append the token very to each example and restrict the language model's output distribution to only the words positive and negative and guess the token it assigns higher probability to as the prediction. Table 2 details various results on the different NLI tasks for our model and previous state-of-the-art approaches. In this paper, we explore a semi-supervised approach for language understanding tasks using a combination of unsupervised pre-training and supervised fine-tuning. We also achieve an overall score of 72.8 on the GLUE benchmark, which is significantly better than the previous best of 68.9.6 Table 4: Semantic similarity and classification results, comparing our model with current state-of-theart methods. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 777–789. (4) (x,y) We additionally found that including language modeling as an auxiliary objective to the fine-tuning helped learning by (a) improving generalization of the supervised model, and (b) accelerating convergence. In our experiments, we use a multi-layer Transformer decoder [34] for the language model, which is a variant of the transformer [62]. For DPRD [46] (winograd schemas), we replace the definite pronoun with the two possible referrents and predict the resolution that the generative model assigns higher average token log-probability to the rest of the sequence after the substitution. λ was set to 0.5.4.2 Supervised fine-tuning We perform experiments on a variety of supervised tasks including natural language inference, question answering, semantic similarity, and text classification. Figure 2(left) illustrates the performance of our approach

on MultiNLI and RACE as a function of the number of layers transferred. As we demonstrate in our experiments, these adaptations enable us to fine-tune effectively with minimal changes to the architecture of the pre-trained model. The closest line of work to ours involves pre-training a neural network using a language modeling objective and then fine-tuning it on a target task with supervision. Further, we also demonstrate the effectiveness of our model on a wider range of tasks including natural language inference, paraphrase detection and story completion. First, we use a language modeling objective on the unlabeled data to learn the initial parameters of a neural network model. We observe that the auxiliary objective helps on the NLI tasks and QQP. We evaluate our approach on four types of language understanding tasks – natural language inference, question answering, semantic similarity, and text classification. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. We observe the performance of these heuristics is stable and steadily increases over training suggesting that generative pretraining supports the learning of a wide variety of task relevant functionality. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding.

Chapter 3

Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models

In this chapter, we propose a novel method to extend sequence-to-sequence models to accurately process sequences much longer than the ones used during training while being sample-and resource-efficient, supported by thorough experimentation [1]. To investigate the effectiveness of our method, we apply it to the task of correcting documents already processed with Optical Character Recognition (OCR) systems using sequence-to-sequence models based on characters. We test our method on nine languages of the ICDAR 2019 competition on post-OCR text correction and achieve a new state-of-the-art performance in five of them. The strategy with the best performance involves splitting the input document in character n-grams and combining their individual corrections into the final output using a voting scheme that is equivalent to an ensemble of a large number of sequence models. We further investigate how to weigh the contributions from each one of the members of this ensemble. Our code for post-OCR correction is shared at `https://github.com/jarobyte91/post_ocr_correction`.

3.1 Introduction

Since its inception in the early sixties, OCR has been a promising and active area of research. Nowadays, systems like Tesseract [74] obtain accuracies above 90% on documents from 19th- and early 20th-century newspaper pages [77], but the accurate recognition of older, historical texts remains an open challenge due to their vocabulary, page layout, and typography. This is why successful OCR systems are language-specific and focus only on resource-rich languages, like English.

As a consequence of these difficulties, the task of automatically detecting and

[1] This chapter is an improved version of the paper [72], which was accepted at the 36th AAAI Conference on Artificial Intelligence (AAAI 2022).

correcting errors in documents has been studied for several decades [47], ranging from techniques based on statistical language modelling [89], dictionary-based translation models [46] or large collections of terms and word sequences [7].

With the advent of methods based on neural networks, and more specifically, sequence models such as [16, 88, 90], the automatic correction of texts using sequence models witnessed considerable progress in the form of neural sequence models based on characters or words [79, 83].

Character-based sequence models offer good generalization due to the flexibility of their vocabulary, but they are challenging to train and inefficient at inference time, as generating a document one character at a time requires thousands of steps. On the other hand, word-based sequence models are efficient at inference time and more sample-efficient than character-based sequence models, but they lack generalization, a problem that has been partially solved with systems like WordPiece [97] or Byte-Pair Encodings [65], that learn useful sub-word units to represent text from the data they are trained on.

In this work, we propose a novel method to correct documents of arbitrary length based on character sequence models. The novelty of our method lies in training a character sequence model on short windows both to detect the mistakes and to generate the candidate corrections at the same time, instead of first finding the mistakes and then use a dictionary or language model to correct them, as is usual with post-OCR text correction systems.

The first main idea behind our method is to use the sequence model to correct n-grams of the document instead of the whole document as a single sequence. In this way, the document can be processed efficiently because the n-grams are corrected in parallel. The other key idea of the method is the combination of all the n-gram corrections into a single output, a process that adds robustness to the technique and is equivalent to using an ensemble of a large number of sequence models, where each one acts on a different segment.

The features that set apart the method proposed in this paper from previous methods for post-OCR text correction are the following:

- It can handle documents of great length and difficulty while being character-based, which means that it can deal with out-of-vocabulary sequences gracefully and be easily applied to various languages.

- It is sample- and resource-efficient, requiring only a couple of hundred corrected documents in some cases to produce good improvements in the quality of the text while needing very modest hardware to train and to perform inference.

- It is robust because it integrates a set of strategies to combine the output of a large ensemble of character sequence models, each one focusing on a different context.

- It sets a new state-of-the-art performance on the ICDAR 2019 competition for post-OCR text correction. The system hereby proposed obtained major improvements in Spanish, German, Dutch, Bulgarian and Czech, while remaining competitive in the remaining languages.

3.2 Related Work

The state of the art in OCR post-processing is reflected in the two editions of the ICDAR competition on Post-OCR text correction [15, 78]. This competition is divided into two tasks: the detection of OCR errors and their correction.

The best performing error detector method during the first edition of the challenge was WFST-PostOCR [59], while the best correction method was Char-SMT/NMT [4]. WFST-PostOCR relies on compiling probabilistic character error models into weighted finite-state edit transducers, while a language model finds the best token sequence. On the other hand, Char-SMT/NMT is based on ensembles of character-based Machine Translation models, each one trained on texts from different periods of time to translate each token within a window of two preceding and one succeeding tokens.

In the second edition of the challenge, the best method for both error detection and correction was Context-based Character Correction (CCC). This method is a fine-tuning of multilingual BERT [28] that applies a machine translation technique based on a character sequence model with an attention mechanism.

The most recent extension to the CCC method also applies BERT and character-level machine translation [60], but it also includes static word embeddings and character embeddings used in a Neural Machine Translation system, and a candidate filter. The method proposed in [82] argues that applying a two-step approach to automatic OCR post-correction reduces both the Character Error Rate (CER) and the proportion of correct characters that were falsely changed. The resulting model consists of a bidirectional LSTM-based detector and a standard LSTM-based sequence-to-sequence translation model.

Unlike CCC, our method does not rely on pre-trained language models, which makes it applicable to low-resource settings without sacrificing performance.

3.3 Methodology

The main idea of our method is to train a sequence model on sequences of characters and then use it to correct complete documents. However, using this approach directly is computationally unfeasible because documents are sequences of thousands of characters, and training a model like this would need an immense amount of both memory and corrected documents. To overcome these limitations, we propose a method composed of three steps, as shown in Fig. 3.1.

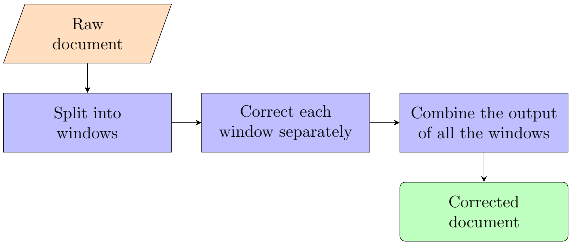
Raw document
Split into windows
Correct each window separately
Combine the output of all the windows
Corrected document

Figure 3.1: Overview of the proposed method. In the first step, the document is split into either disjoint windows or n-grams. In the second step, the windows are corrected in parallel using the sequence model. In the third step, the partial corrections obtained in the previous step are combined to obtain the final output: by a simple concatenation when using disjoint windows or a voting scheme when using n-grams. After the merging step, the final output can be compared with the correct transcription using Character Error Rate.

3.3.1 The Sequence Model

The core of our system is a standard sequence-to-sequence model that can correct sequences of characters. In our implementation, we used a Transformer [90] as the sequence model, which takes as input a segment of characters from the document to correct, and the output is the corrected segment. To train this sequence model, it is necessary to align the raw documents with their corresponding correct transcriptions, which is not always straightforward.

Since the output is not necessarily of the same length as the input (because of possible insertions or deletions of characters), a decoding method like Greedy Search or Beam Search is needed to produce the most likely corrected sequence according to the model.

3.3.2 Processing Full Documents with the Sequence Model at Inference Time

Assuming that the sequence model is already trained, the next step is to use it to correct texts of arbitrary length. This can be done by splitting the document into windows with a length similar to the ones on which the model was trained and combining them with the strategies we describe next.

Disjoint Windows

Correcting a document by splitting it into disjoint windows is the most basic way to use the sequence model to process a string that is longer than the maximum sequence it allows. In the splitting step, the string to correct is split into disjoint windows of a fixed length n. In the correction step, each window is corrected in parallel using the sequence model. In the merging step, the final output is produced by concatenating the corrected output from each window. To evaluate the method, the final output can be compared with the correct transcription using the CER.

It is important to note that this approach can be effective if the sequence model is well trained, but if this is not the case, it can be prone to a *boundary effect*, where the characters at the ends of the windows do not have the appropriate context. An example of this approach is shown in Fig. 3.2.

N-Grams

To counter the *boundary effect*, it is possible to add robustness to the output by using all the n-grams of the input. In the splitting step, the string to correct is split into character n-grams. In the correction step, each window is corrected in parallel using

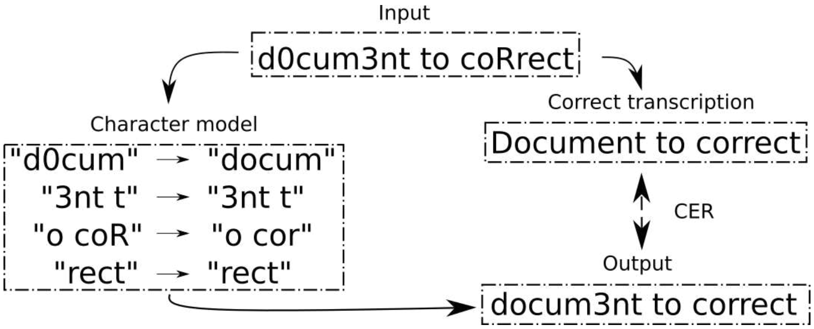

Figure 3.2: An example of correcting a document using disjoint windows of length 5.

the sequence model. The merging step produces the final output by combining the output from the windows, taking advantage of the overlapping between them and a voting scheme influenced by a weighting function described below. To evaluate the method, the final output is compared with the correct transcription using the CER. An example of this method is depicted in Fig. 3.3.

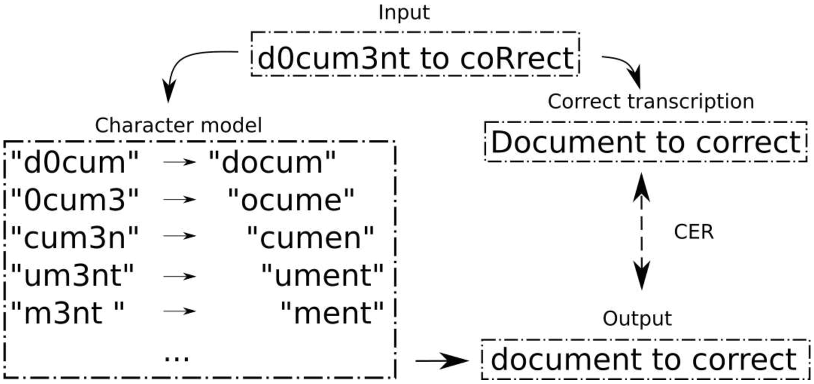

Figure 3.3: An example of correcting a document using n-grams of length 5.

An essential part of the n-grams variation is how the partial outputs are combined. Since the partial corrections have an offset of one, the outputs can be combined by aligning them and performing a vote to obtain the most likely character for every position. This vote is equivalent to processing the whole input with an ensemble of n models, each one operating on segments of offset 1, where n is the order of the n-grams.

Since a character corrected in the middle of an n-gram has more context than a character in the edges, it is reasonable to think that they should have different weights in the vote. To express this difference, we used three different weighting functions, given by the following formulae:

$$bell(p, w) = exp\left(-\left(1 - \frac{p}{m}\right)^2\right),$$

$$triangle(p, w) = 1 - \frac{|m - p|}{2m},$$

$$uniform(p, w) = 1,$$

where p is the character position in the window, w is the window length, and $m = \lceil \frac{w}{2} \rceil$.

The weight of the character vote in position p in an n-gram of length w is given by $f(p, w)$, where f is one of the weighting functions. An example of this is shown in Fig. 3.4.

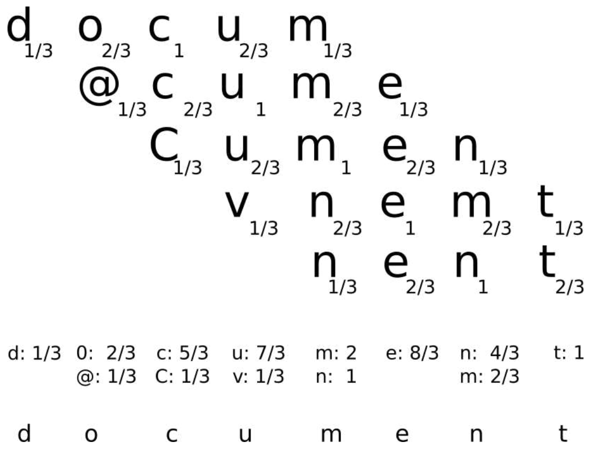
d 1/3 o 2/3 c 1 u 2/3 m 1/3
@ 1/3 c 2/3 u 1 m 2/3 e 1/3
C 1/3 u 2/3 m 1 e 2/3 n 1/3
v 1/3 n 2/3 e 1 m 2/3 t 1/3
n 1/3 e 2/3 n 1 t 2/3
d: 1/3
0: 2/3
@: 1/3
c: 5/3
C: 1/3
u: 7/3
v: 1/3
m: 2
n: 1
e: 8/3
n: 4/3
m: 2/3
t: 1
d o c u m e n t

Figure 3.4: An example of correcting a text with 5-grams and the *triangle* weighting function. The number under every character in the top part is the weight of that character in its position for every window. The mid-bottom table shows the sum of the weights for every candidate character on each position of the output. To generate the final output (at the bottom), the candidate character with the maximum sum on every position is selected.

3.4 Experimental Setup

3.4.1 Data

The dataset of the ICDAR2019 Competition on Post-OCR Text Correction is made of 14,309 documents scanned with OCR along with their corresponding correct transcription in 10 languages: Bulgarian (bg), Czech (cz), German (de), English (en), Spanish (es), Finnish (fi), French (fr), Dutch (nl), Polish (pl) and Slovak (sl). In this work, we used all of them except for Finnish because the files required are distributed separately due to copyright. The details of the datasets used are shown in Table 3.1.

Language	Total documents	μ length	μ CER	σ CER	Train documents	Best % improvement
bg	198	2,332	16.65	16.30	149	9.0
cz	195	1,650	5.99	12.98	149	6.0
de	10,080	1,546	24.57	5.86	8,052	24.0
en	196	1,389	22.76	23.81	148	11.0
es	197	2,876	31.52	22.65	147	11.0
fr	2,849	1,521	8.79	12.15	2,257	26.0
nl	198	4,289	28.11	25.00	149	12.0
pl	199	1,688	36.68	20.50	149	17.0
sl	197	1,538	12.50	19.85	149	14.0

Table 3.1: The ICDAR datasets. μ length is the average document length measured in characters. μ CER and μ CER are the mean and standard deviation of the Character Error Rate between every document and its correct transcription. Best % improvement is the percentage of improvement in the CER from the best method reported in [78].

3.4.2 Obtaining Sequence Pairs for the Sequence Model

To obtain the sequences to train the sequence model, the format of the ICDAR datasets was crucial. The alignment process we followed is described in Fig. 3.5.

To create a development set for each language, we sampled five documents from each training set and then split the ground truth of every document into n-grams of length 100 to create the input-correction pairs to train and develop the sequence models. We chose this number of documents to be able to evaluate the models frequently and this length because this was the largest one that fitted in our hardware with the largest architectures we tried. The datasets used to train our models are described in Table 3.2.

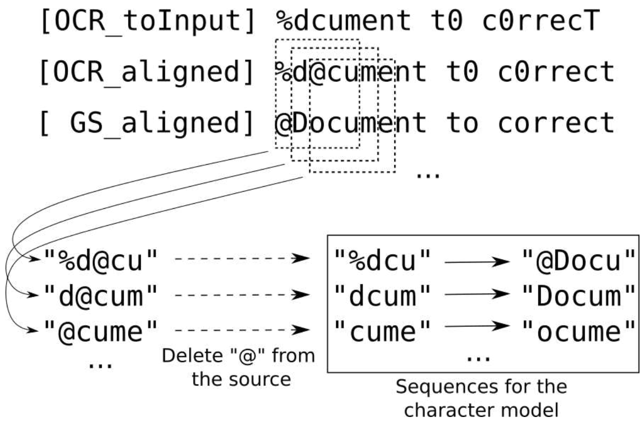
[OCR_toInput] %dcument t0 c0rrecT
[OCR_aligned] %d@cument t0 c0rrect
[GS_aligned] @Document to correct
...
"%d@cu"
"d@cum"
"@cume"
...
Delete "@" from the source
"%dcu"
"dcum"
"cume"
"@Docu"
"Docum"
"ocume"
...
Sequences for the character model

Figure 3.5: An example of the process to train the sequence model using the ICDAR datasets with windows of length 5. In the first step, the correct transcription of the document (*GS_aligned*) is split into n-grams, and for each one, the corresponding part of the aligned input (*OCR_aligned*) is retrieved. In the second step, the character @ is deleted only from the aligned input to obtain a set of segments from the document (*OCR_toInput*) paired with their correction.

Language	Train			Development		
	μ length	μ CER	Pairs	μ length	μ CER	Pairs
bg	1,872	16.14	278.3	1,708	9.39	8.7
cz	1,638	6.02	238.3	2,017	10.56	10.1
de	1,547	24.52	12,779.5	1,531	22.84	7.7
en	1,419	23.83	217.9	1,295	45.62	7.3
es	2,967	30.84	466.2	2,110	43.40	11.4
fr	1,534	8.63	3,553.8	1,643	5.53	4.9
nl	4,293	28.38	666.6	3,762	32.62	21.5
pl	1,666	40.08	259.6	1,463	29.95	7.8
sl	1,383	11.24	208.2	1,457	1.25	7.3

Table 3.2: The datasets used to train the sequence models. μ length is the average character length of the documents. μ CER is the average Character Error Rate between each document and its correct transcription. Pairs is the number in thousands of segment-correction pairs obtained.

3.4.3 Training the Sequence Models

The process of training the models is the standard sequence-to-sequence pipeline that uses cross entropy loss to make the model generate the right token at every step, as it was proposed in [16, 88]. All the models were trained using 4 CPU cores, 4 GB of RAM, and a single GPU NVIDIA V100 with 16 GB of memory. Overall, training the sequence models was difficult because of the differences between the training and development sets, but the models obtained were good enough to produce improvements in all the languages, as shown in Table 3.3.

Language	Best epoch	Total epochs	Dev loss	Train loss	Parameters	Train hours
bg	19	42	0.278	0.251	1.94	2.19
cz	2	50	0.255	0.095	15.05	3.65
de	7	7	0.330	0.406	2.00	1.93
en	25	50	1.010	0.455	3.84	1.52
es	19	24	1.077	0.688	3.86	1.61
fr	10	12	0.318	0.288	1.48	1.88
nl	8	16	0.583	0.468	7.54	2.97
pl	10	47	0.594	0.578	7.56	3.41
sl	15	57	0.035	0.157	3.82	1.78

Table 3.3: Training of the models. *Best epoch* is the epoch with the lowest dev loss. *Dev loss* is the lowest loss on the dev set. *Train loss* is the loss on the train set in the best epoch. *Parameters* is the model parameters in millions.

To tune the hyper-parameters of the sequence models, we performed a Random Search [39]. We set the embedding dimension to be 128, 256, or 512, with the number of hidden units in the feed-forward layers always four times the embedding dimension. We tried from two to four layers, with the same number of layers for both the encoder and the decoder. We varied the dropout rate from 0.1 to 0.5 in steps of 0.1 and the λ of the weight decay L^2 penalization to be 10^{-1}, 10^{-2}, 10^{-3} or 10^{-4}. All the models were trained with Adam and a learning rate of 10^{-4}. The best hyper-parameters found are shown in Table 3.4.

Language	Embedding Dimension	Feed-Forward Dimension	Layers	Dropout
bg	128	512	4	0.2
cz	512	2,048	2	0.1
de	128	512	4	0.3
en	256	1,024	2	0.5
es	256	1,024	2	0.4
fr	128	512	3	0.5
nl	256	1,024	4	0.2
pl	256	1,024	4	0.3
sl	256	1,024	2	0.2

Table 3.4: Hyper-parameters of the sequence models. All the models were trained with Adam[44], a learning rate of 10^{-4} and a weight decay L^2 penalization of 10^{-4}.

3.4.4 Experimental Results

To investigate the effect of the different hyper-parameters in our method, we performed a Grid Search varying the window size from 10 to 100 in steps of 10, processing the documents with disjoint windows or n-grams with all the weighting functions using both Greedy Search and Beam Search.

The best model for each language is shown in Table 3.5. The effect of each one of the hyper-parameters (window type, decoding method, weighting function and window size) in the average improvement in CER is shown in Tables 3.6, 3.7, 3.8 and 3.9. The best model in CER obtained for every combination of language and window size is shown in Table 3.10. The average percentage of improvement in CER for every combination of language, window type, decoding method, and weighting function is shown in Table 3.11. The average inference time for every combination of language, window type, decoding method, and weighting function is shown in Table 3.12.

Language	Window type	Window size	Decoding method	Weighting function	Inference time	μ CER before	μ CER after	% Improvement	% Baseline
bg	N-grams	80	Beam	Uniform	198.08	18.23	15.27	**16.27**	9.0
cz	N-grams	40	Beam	Uniform	37.90	5.90	4.52	**23.36**	6.0
de	N-grams	100	Beam	Triangle	4,340.23	24.77	15.62	**36.94**	24.0
en	N-grams	20	Beam	Uniform	10.37	19.47	18.00	7.52	11.0
es	N-grams	60	Beam	Triangle	70.58	33.54	29.41	**12.30**	11.0
fr	N-grams	90	Beam	Triangle	889.27	9.40	7.88	16.18	26.0
nl	N-grams	80	Greedy	Uniform	47.35	27.30	22.41	**17.94**	12.0
pl	N-grams	10	Greedy	Uniform	1.68	26.56	23.19	12.69	17.0
sl	N-grams	90	Beam	Uniform	85.73	16.42	14.64	10.83	14.0
					Average	20.17	16.77	17.11	14.4

Table 3.5: Best approach found for every language on the ICDAR test sets. *μ CER before* and *μ CER after* is the average Character Error Rate between every document and its correct transcription, before and after using our method. *% Improvement* is the average percentage of improvement in CER. *% Baseline* is the average percentage of improvement in CER from the best method in [78].

Window Type	Mean	Std	Min	25%	50%	75%	Max
Disjoint	-6.83	65.48	-422.21	-2.21	4.12	11.63	36.12
N-grams	0.11	67.31	-423.77	6.10	10.82	16.67	36.94

Table 3.6: Descriptive statistics of the average percentage of improvement in CER on the ICDAR test sets grouped by window type.

Decoding Method	Mean	Std	Min	25%	50%	75%	Max
Beam Search	-6.33	79.12	-423.77	3.69	9.74	16.07	36.94
Greedy Search	3.09	51.51	-403.78	5.48	9.03	16.06	35.20

Table 3.7: Descriptive statistics of the average percentage of improvement in CER on the ICDAR test sets grouped by decoding method.

Weighting Function	Mean	Std	Min	25%	50%	75%	Max
Bell	-0.10	67.47	-423.76	5.94	10.43	16.36	36.89
Triangle	0.03	67.50	-423.77	6.06	10.56	16.58	36.94
Uniform	0.41	67.32	-423.76	6.38	10.92	16.77	36.83

Table 3.8: Descriptive statistics of the average percentage of improvement in CER on the ICDAR test sets grouped by weighting function.

Window Size	Mean	Std	Min	25%	50%	75%	Max
10	-36.93	132.72	-423.73	2.27	5.39	12.48	31.70
20	-25.77	112.54	-423.77	4.74	8.00	14.58	33.22
30	-16.15	96.76	-408.21	4.64	8.64	15.48	33.79
40	3.47	34.78	-156.62	5.36	8.45	16.55	34.59
50	11.37	11.59	-21.38	6.10	10.00	17.08	36.12
60	11.72	11.33	-25.74	6.53	11.96	17.15	36.16
70	12.43	10.97	-21.27	6.68	12.46	16.30	36.19
80	11.89	12.34	-29.57	6.61	11.78	16.82	36.56
90	8.06	15.71	-47.30	-1.39	10.41	15.75	36.63
100	3.70	26.46	-93.51	-7.47	9.01	16.74	36.94

Table 3.9: Descriptive statistics of the average percentage of improvement in CER on the ICDAR test sets grouped by window size.

Language	Window Size									
	10	20	30	40	50	60	70	80	90	100
bg	-366.79	-229.28	-63.67	4.40	13.01	14.60	16.02	**16.27**	15.77	15.42
cz	16.66	20.80	21.81	**23.36**	22.33	18.49	19.76	21.61	15.84	22.02
de	31.70	33.22	33.79	34.59	36.12	36.16	36.19	36.56	36.63	**36.94**
en	5.45	**7.52**	6.88	7.10	6.26	6.13	4.75	2.31	-1.06	-7.03
es	4.82	8.00	9.35	11.00	12.03	**12.30**	11.91	11.79	10.92	9.14
fr	8.47	10.93	11.50	11.80	13.44	14.68	15.34	15.81	**16.18**	16.07
nl	14.35	15.94	16.50	17.10	17.45	17.49	17.82	**17.94**	17.73	17.14
pl	**12.69**	12.47	10.48	9.45	7.21	7.27	7.39	8.55	8.31	8.62
sl	5.46	6.20	6.44	6.97	7.95	9.84	10.08	10.23	**10.83**	9.24

Table 3.10: Best improvement in CER obtained for every language and for every window size on the ICDAR test sets. The best model for every language is bolded.

Language	Disjoint		N-Grams					
	Beam	Greedy	Beam			Greedy		
			Bell	Triangle	Uniform	Bell	Triangle	Uniform
bg	-134.21	-71.40	-129.94	-129.87	-129.26	-61.10	-60.93	-59.86
cz	14.73	13.61	19.50	19.67	19.91	19.17	19.32	19.43
de	33.13	31.21	35.11	35.13	34.97	33.33	33.36	33.20
en	-3.06	-3.16	1.81	1.90	2.14	2.96	3.05	3.22
es	3.37	4.58	6.75	6.79	6.90	7.71	7.73	7.75
fr	9.61	2.03	12.68	12.84	13.39	11.14	11.32	11.93
nl	4.54	7.10	13.89	14.02	14.52	15.99	16.10	16.41
pl	-26.01	-1.39	-12.21	-11.92	-10.66	8.24	8.51	9.24
sl	-2.12	-5.50	7.74	7.92	8.30	5.41	5.53	5.94

Table 3.11: Average percentage of improvement of CER by language for each variation of our method on the ICDAR test sets.

Language	Disjoint		N-Grams					
	Beam	Greedy	Beam			Greedy		
			Bell	Triangle	Uniform	Bell	Triangle	Uniform
bg	3.42	0.84	269.11	270.16	275.63	38.73	38.55	38.11
cz	1.88	0.49	160.87	161.15	160.77	30.49	30.54	30.54
de	61.77	22.78	4,489.84	4,340.23	4,372.81	606.48	602.05	612.37
en	0.93	0.39	66.70	66.29	64.41	10.51	10.53	10.55
es	1.79	0.46	149.66	149.99	148.28	23.23	23.31	23.31
fr	13.38	6.14	1,617.90	934.24	932.64	127.17	127.33	127.40
nl	4.53	1.03	443.59	422.70	424.34	70.29	70.38	69.87
pl	2.03	0.72	175.27	172.24	166.71	28.53	28.50	28.48
sl	1.30	0.42	101.95	102.48	100.76	16.27	16.35	16.30

Table 3.12: Average inference time in minutes for every language and every variation of our method on the ICDAR test sets.

3.5 Discussion

Our method outperformed the state of the art in Bulgarian (bg), Czech (cz), German (de), Spanish (es), and Dutch (nl), while exhibiting comparable performance in the remaining languages, as shown in Table 3.5. The results obtained are interesting for several reasons:

- The method was not as effective in French as it was in German, the other language with abundant training data.

- The choice of weighting function did not have much impact on the performance, although broadly speaking, the best weighting function was *uniform*. Although counter-intuitive, this results means that what matters the most for the method is the number of windows that agree for a given correction, as opposed to their position inside the window.

- Although the method is stable with respect to changes in the window size, a larger window size does not always lead to improved performance. It can sometimes hurt the model's performance, a behavior that appears to be language-dependent, as in the case of English and Polish, according to Table 3.11.

- Although the best results were consistently obtained with Beam Search, Greedy Search seems to be a safer choice than Beam Search. Using Beam Search is between three and ten times slower than using Greedy Search, but these extra computations are usually not justified given that there is no guarantee of increased performance, and even when the performance does increase, the difference is small, as shown in Tables 3.11 and 3.12.

It is important to note that the datasets come from several heterogeneous sources with varying levels of quality and content. In the French dataset, we noticed two important properties: a large portion of the documents are receipts, with little to no narrative text, while the longest documents have very few errors, therefore not allowing much room for improvement, as shown in Fig. 3.6.

After informal manual inspection of the testing sets, we observed that the French model mostly learned to discard parts of the document and to correct numbers and dates. On the other hand, the German model learned to correct the narrative parts. It is important to also note that most models in the original competition also performed poorly in French, while those with the best performance in French used external resources such as Google Book N-grams [78].

3.6 Conclusions and Future Work

The method proposed in this paper allows processing very long texts using character sequence-to-sequence models, which makes it applicable to any language. The method

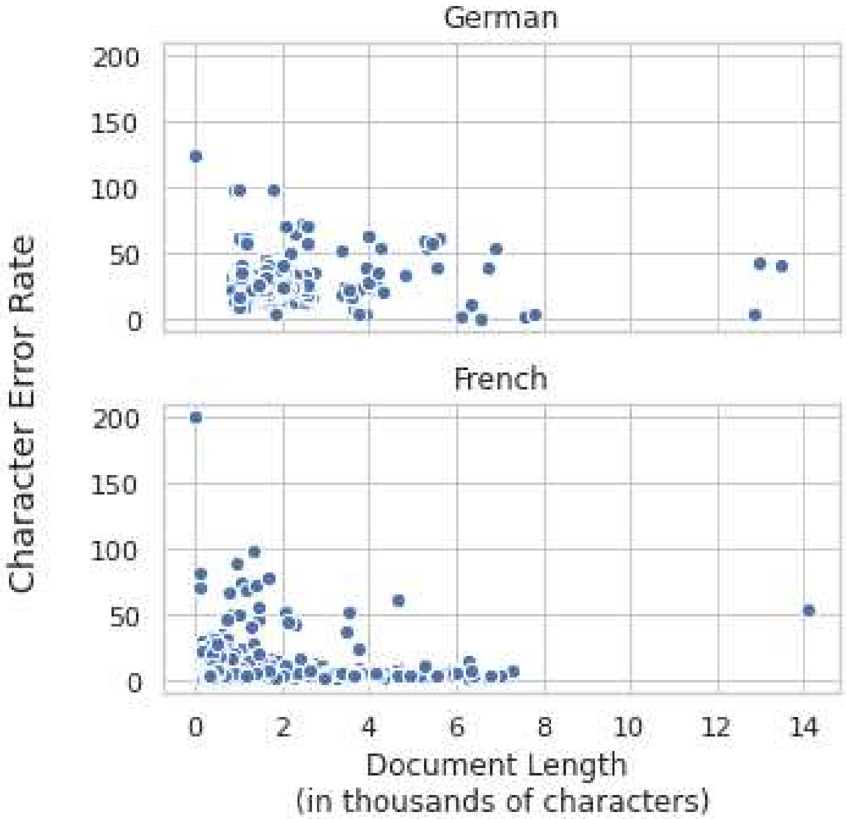

Figure 3.6: Distribution of the length in characters against the Character Error Rate for each document in the German and French datasets.

is simple, resource-efficient and easily parallelizable, obtaining from modest to very good improvements in documents of varying length and difficulty.

Although this paper is focused on text and post-OCR correction, the methods presented here can be transferred to many other sequence problems that require only local dependencies to be solved successfully, requiring very modest hardware and just a couple hundred examples in some cases.

For future work, it would be interesting to apply this method to text from Automated Speech Recognition or Handwritten Text Recognition systems, but the problem of aligning the system's output with the correct transcription remains.

Chapter 4

QuOTeS: Query-Oriented Technical Summarization

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 [1]. 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`.

4.1 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.

[1] This chapter is an improved version of the paper [73], accepted at the 17th International Conference on Document Analysis and Recognition (ICDAR 2023)

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. 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 [42], 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 [63], ROUGE [54], and METEOR [6] 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 [81], they often cannot satisfy their users' needs, given the inherent difficulty and ambiguity of the task [25]. An alternative approach to make systems more user-centric is Query-Focused Summarization [25], 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 [50], 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 [25] and High-Recall Information Retrieval [22] to provide Interactive Query-Focused Summarization of scientific documents. An overview of how *QuOTeS* works and its components is shown in Fig. 4.1.

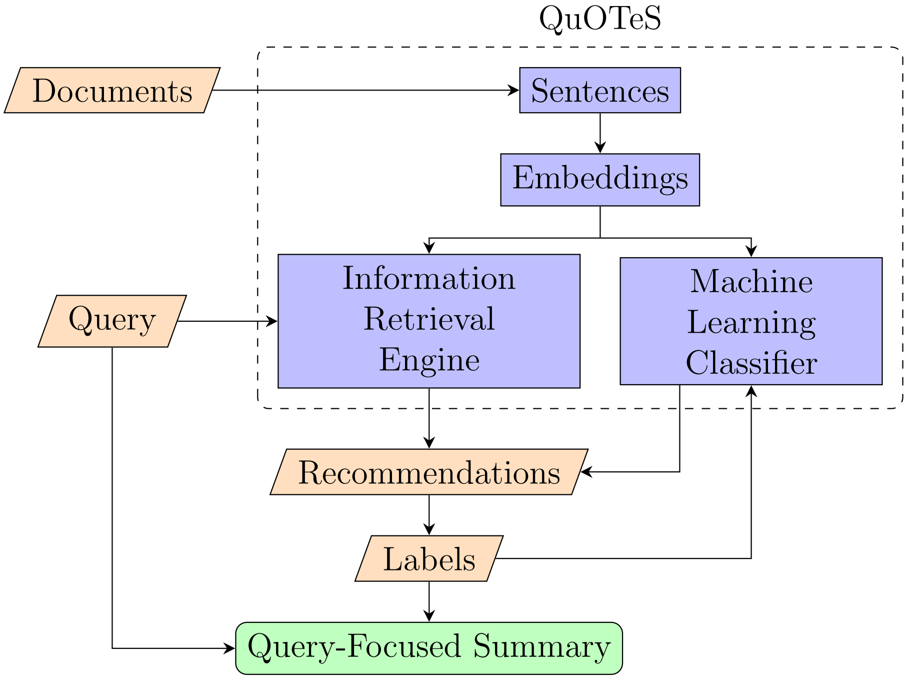

Figure 4.1: Overview of how *QuOTeS* works. First, the user inputs their documents into the system, which then extracts the text present in them. Next, the system splits the text into sentences and computes an embedding for each one of them. After that, the user inputs their query, which is a short paragraph 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.

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 in the collection. The system extracts the text directly from the academic documents provided by the user at run-time, 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 [12] and custom questionnaires to measure its usability and the quality of the summaries it produces.

4.2 Related Work

4.2.1 Query-Focused Summarization

The task of Query-Focused Summarization (QFS) was introduced in the 2005 Document Understanding Conference (DUC 2005) [25]. 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 [26] and DUC 2007 [27]) 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 [8]. 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 summarization systems. 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 [85]. 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.

4.2.2 High-Recall Information Retrieval

A novel evaluation toolkit that simulates a human reviewer in the loop was introduced in [22]. 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 [23]. Through simulations using Cormack and Grossman's Technology-Assisted Review Evaluation Toolkit [22], 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 [24]. This novel variation requires $O(log(N))$ labeling effort and $O(Nlog(N))$ 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.

4.2.3 Interactive Query-Focused Summarization

A novel system that provides summaries for Computer Science publications was introduced in [29]. 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 [99]. 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 [9]. 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.

4.3 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 [85]. 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 run-time. 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 [22] as the training procedure for the classifier inside *QuOTeS*.

4.4 System Design

QuOTeS is a browser-based interactive system built with *Python*, mainly using the *Dash* package [66]. 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. 4.2.

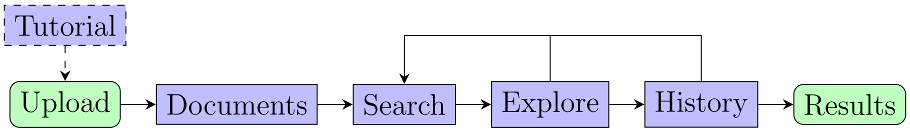

Figure 4.2: Methodology of the system and its workflow.

4.4.1 *Tutorial*

In this step, the user can watch a 5-minute video[1] 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.

4.4.2 *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 tri-grams and Sentence-BERT embeddings [75]. 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.

4.4.3 *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`

4.4.4 *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 five recommendations.

4.4.5 *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 the batch of five recommendations 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.

4.4.6 *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.

4.4.7 *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.

4.5 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 [62], 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 4.3.

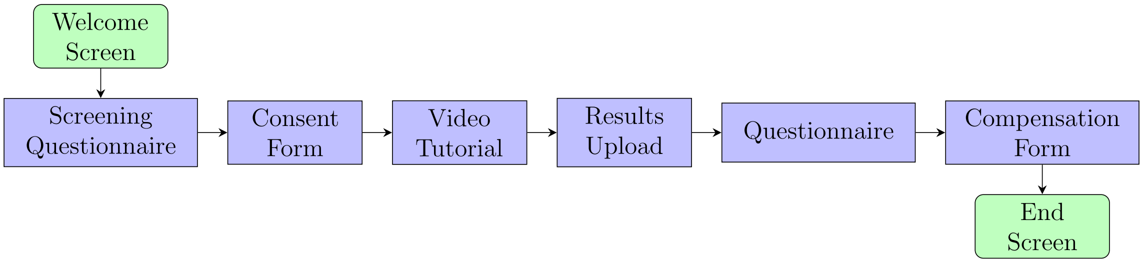

Figure 4.3: Overview of the user study.

4.5.1 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.

4.5.2 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 4.1, while the full results of the screening questionnaire can be found in the code repository.

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

Table 4.1: Responses of the Screening Questionnaire from the participants that completed the study.

4.5.3 Research Instrument

During the user study, the participants filled out a questionnaire composed of thirty questions divided into three parts: *Usability*, *Features*, and *Summary Quality*. In the *Usability* part, they filled out the questionnaire from the standard *System Usability Scale* [12], 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 Section 4.8.

4.5.4 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. 4.4.

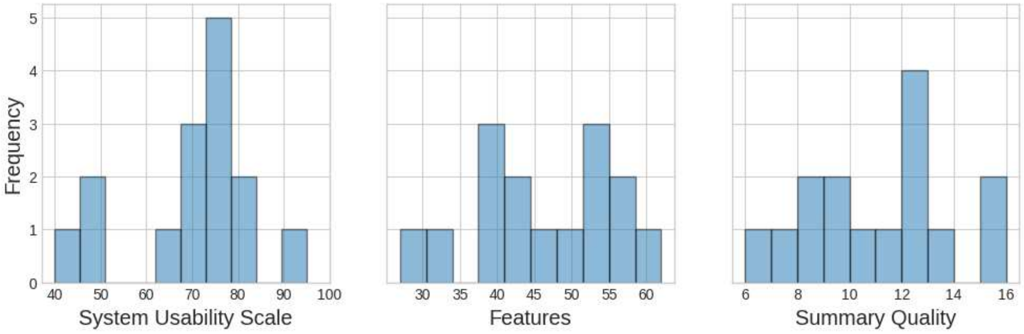

Figure 4.4: Distribution of the questionnaire scores obtained during the user study. The possible range for each one of the scores is the following: *System Usability Scale* ranges from 0 to 100, with a mean of 69.67 and a median of 75; the *Features* score ranges from 0 to 64 with a mean of 45.87 and 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.

4.6 Discussion

4.6.1 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:

- 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 uni-grams, character tri-grams, 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. 4.5. 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.

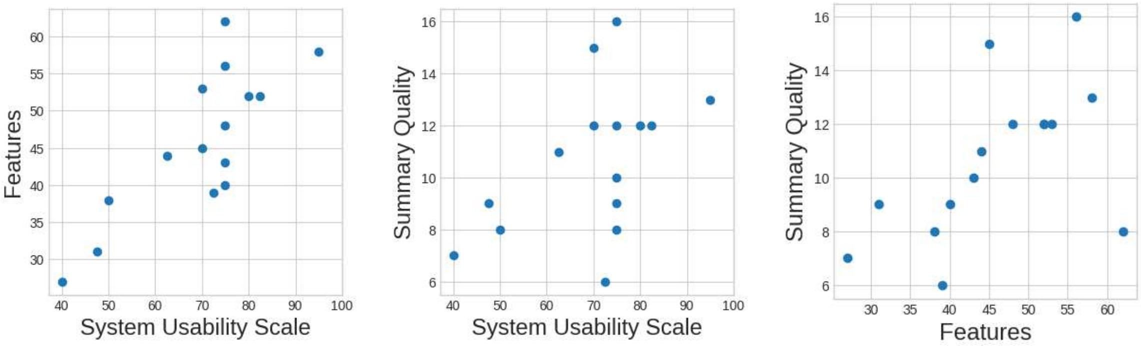

Figure 4.5: Relationship between the scores computed from the questionnaires.

4.6.2 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 [8], using the Kullback-Leibler Divergence [48] between the uni-gram vocabulary of the document collection and the uni-gram vocabulary of the query-focused summary produced.

The distributions of the Precision and KL-Divergence, along with their relationship, are shown in Fig. 4.6. 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.

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. 4.7. 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

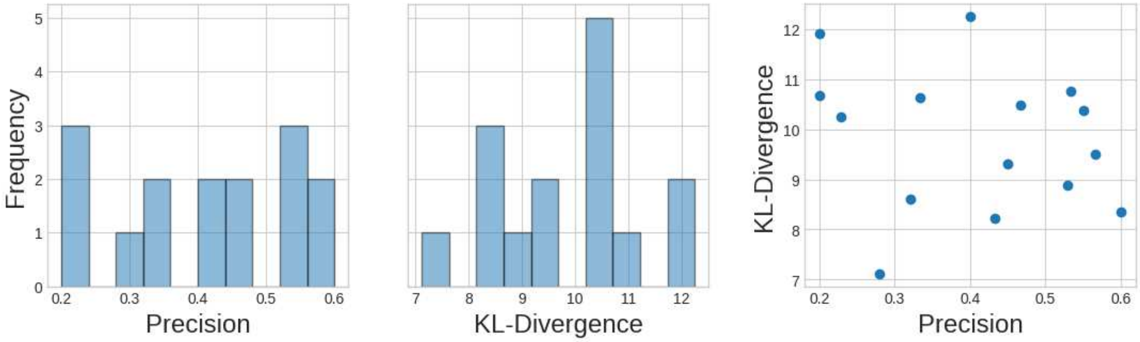

Figure 4.6: Distributions of the Precision of the system (left) and the Kullback-Leibler Divergence between the word uni-gram distribution of the document collections and the summaries produced (center), along with their relationship (right).

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.

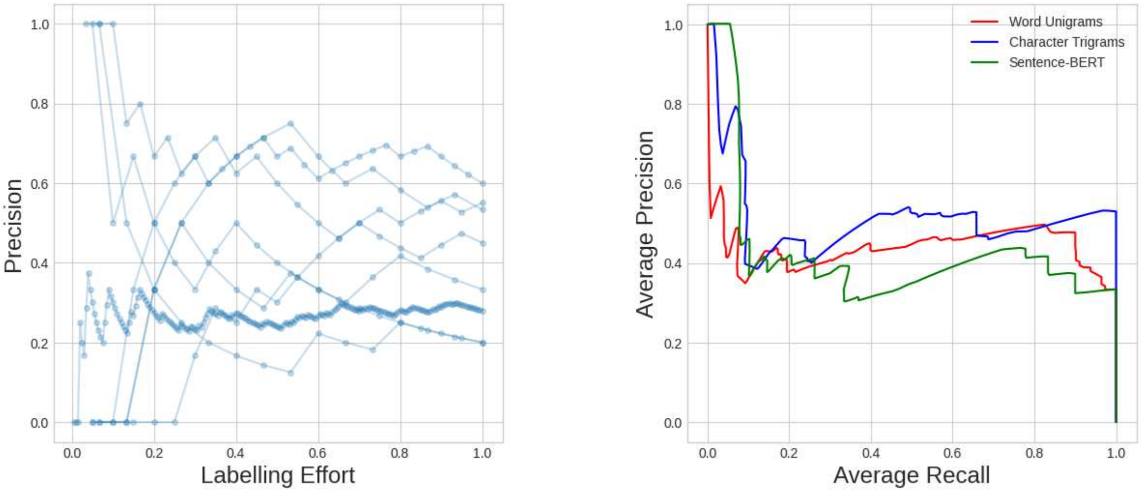

Figure 4.7: Precision of the system. Precision as a function of the Labeling Effort for each one of the participants in the user study (left). Average Precision-Recall Curve of the different embeddings after removing the interactive component of *QuOTeS* (right).

The relationships between the Precision and the scores obtained from the questionnaires in the user study are shown in Fig. 4.8. 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. 4.6. 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.

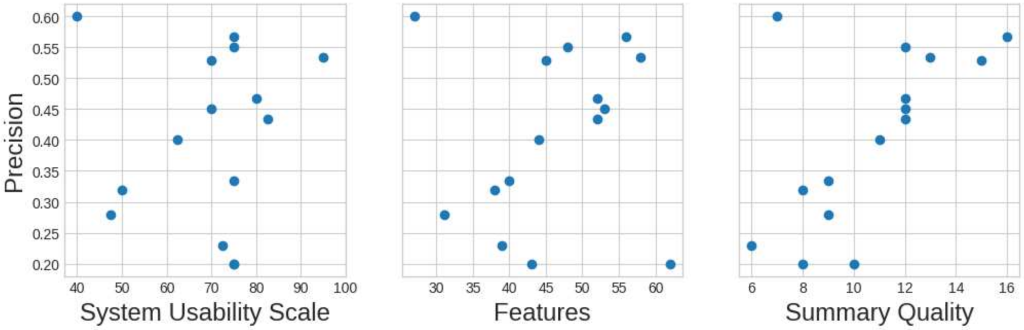

Figure 4.8: Relation between the Precision of the system and the questionnaire scores.

The relationship between the Divergence and the scores is shown in Fig. 4.9. 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.

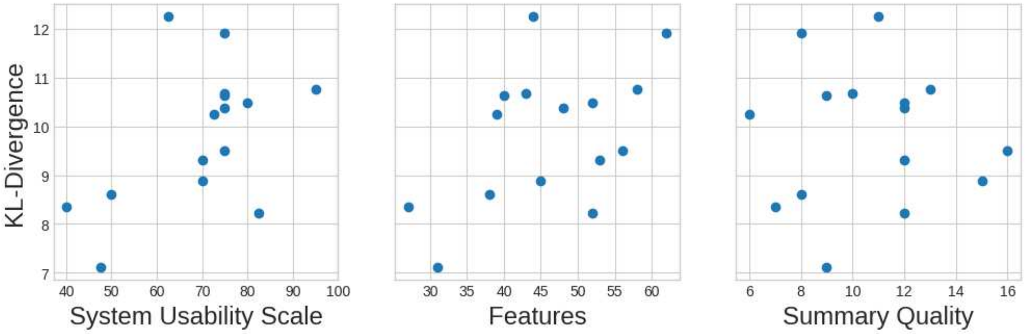

Figure 4.9: Relationship between the Kullback-Leibler Divergence between the word uni-gram distribution of the document collection and produced summaries versus the questionnaire scores obtained in the user study.

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. 4.1). We evaluated the three variations of the baseline system as they appear inside *QuOTeS*. The performance obtained by this baseline is shown in Fig. 4.7.

Even when using Sentence-BERT embeddings, the performance of the baseline system is markedly inferior compared to that of *QuOTeS*, as shown in Fig. 4.7. 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.

4.6.3 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.

4.7 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.

4.8 Questionnaires

All questions in the questionnaire were measured in a Likert Scale [53] with five levels: *Strongly Disagree*, *Somewhat Disagree*, *Neutral*, *Somewhat Agree* and *Strongly Agree*.

Question	StD	soD	N	soA	StA
01. I think I would like to use this system frequently	0	1	2	6	6
02. I found the system unnecessarily complex	4	5	5	1	0
03. I thought the system was easy to use	0	1	2	6	6
04. I think that I would need the support of a technical person to be able to use this system	5	7	2	0	1
05. I found the various functions in this system were well integrated	1	1	1	6	6
06. I thought there was too much inconsistency in this system	4	6	3	1	1
07. I would imagine that most people would learn to use this system very quickly	0	3	0	8	4
08. I found the system very cumbersome to use	5	4	2	3	1
09. I felt very confident using the system	1	1	2	5	6
10. I needed to learn a lot of things before I could get going with this system	4	2	2	5	2

Table 4.2: Results of the System Usability Scale (SUS) Questionnaire.

Question	StD	soD	N	soA	StA
11. It was completely clear what the system does and how it works	0	2	0	7	6
12. The instructions for the task were very difficult to understand	6	4	2	3	0
13. The tutorial told me absolutely everything I needed to know about the system and how to use it	1	1	2	2	9
14. The effect of the settings was very difficult to understand	5	1	5	3	1
15. I completely understood the purpose of the system	0	1	0	5	9
16. It was very hard to decide if the sentences are related to the query	2	5	1	7	0
17. The system is too slow to be usable	2	5	1	6	1
18. The system has all the features needed to perform the task	1	2	4	5	3
19. There are features for which I don't understand the purpose	7	1	0	5	2
20. The *Documents* tab is useful	0	0	2	6	7
21. The *Search* tab is useful	0	3	0	7	5
22. The *Explore* tab is useful	0	3	0	3	9
23. The *History* tab is useful	0	0	1	6	8
24. The *Results* tab is useful	0	1	0	6	8
25. The plots in the system are very hard to understand	4	3	2	4	2
26. I found the information presented in the plots very useful	0	2	3	6	4

Table 4.3: Results of the System Features Questionnaire.

Question	StD	soD	N	soA	StA
27. I think that the sentences recommended by the system are relevant most of the time	1	0	3	7	4
28. I think that the summaries produced by the system are redundant most of the time	3	5	5	1	1
29. I think that the summaries produced by the system are concise most of the time	0	2	4	6	3
30. I think that the summaries produced by the system are incomplete most of the time	2	7	2	3	1

Table 4.4: Results of the Summary Quality Questionnaire.

4.9 Appendix: System Output Examples

1. **Query:** Obesity is a significant problem in populations worldwide, affecting all age groups alike. According to the World Health Organization (WHO) website (2021), around 39% of the world population of adults aged 18 years and over were overweight in 2016, and 13% were obese. In 2019, over 340 million children and adolescents aged 5-19 were overweight or obese (WHO, 2021). The majority of the world's population today lives in nations where obesity and overweight kill more people than underweight (World Obesity, 2022). However, this is preventable if underlying factors leading to weight gain are identified and

precautionary measures are taken to avoid being overweight and obese. In this study, factors were identified that have direct influence on Obesity in Males and Females separately, and individuals were then classified according to the response variable 'Obesity' into seven distinct levels, namely, Insufficient Weight, Normal Weight, Overweight Levels I, II and Obesity Levels I, II and III, with Obesity Level III being morbidly obese. The study used supervised learning techniques such as Logistic Regression (One vs. Rest approach), Decision Tree and Random Forest on data collected from South American countries of Chile, Peru, and Mexico; the highest performance was achieved in the Random Forest algorithm with an accuracy of 96.55%. **Query-Focused Summary:** Keywords: Obesity, Data Mining, Semma, Decision Trees, Naive Bayes, Logistic Regression, Weka, Java Introduction The World Health Organization (WHO) (OMS, 2016), describes obesity and overweight as excessive fat accumulation in certain body areas that can be harmful for health, the number of people that suffers from obesity has doubled since 1980 and also in 2014 more than 1900 million adults, 18 years old or older, are suffering from alteration of their weight. Once the dataset was validated and prepared, the data mining techniques and methods were applied, using the Weka tool, that has a set of algorithms that can be applied to many situations. WEKA is able to support many data mining activities to forecast health problems, such as data preprocessing, classification, grouping, simulation, correlation, and functional choice. Finally, a software was built to use and train the selected method, using the Weka library. To be able to use the data mining methods, we added the Weka Toolkit (weka.jar), in Fig. 4 you can see the library import in the tool used for it. The class level precision, evaluation method and the data analysis results rely on WEKA's software using different machine learning algorithms. Optimization Strategy In order to enhance the classification results and to obtain accuracy-based better performance, the Weka meta-learner (CV Parameter Selection) search methodology was used [27]. Using WEKA, the Decision Trees technique was observed to have the best precision rate of 97.4%. Next, three techniques, Bayesian networks, Logistic Regression, and Decision trees, were chosen.

2. **Query:** Current methods of assessing dementia Alzheimer type (DAT) in older

adults involve structured in- terviews that attempt to capture the complex nature of deficits suffered. One of the most significant areas affected by the disease is the capacity for functional communication as linguistic skills break down. These methods often do note capture the true nature of language deficits in spontaneous speech. We address this issue by exploring novel automatic and objective methods for diagnosing patients through analysis of spontaneous speech. We detail several lexical approaches to the problem of detecting and rating DAT. The approaches explored rely on character n-gram-based techniques, shown recently to perform successfully in a different, but related task of automatic au- thorship attribution. We also explore the correlation of usage frequency of different parts of speech and DAT. We achieve a high 95% accuracy of detecting dementia when compared with a control group, and we achieve 70% accuracy in rating dementia in two classes, and 50% accuracy in rating dementia into four classes. Our results show that purely computational solutions offer a viable alternative to standard approaches to diagnosing the level of impairment in patients. These results are significant step forward toward automatic and objective means to identifying early symptoms of DAT in older adults. **Query-Focused Summary:** Participating teams built language topic models (e.g. an anxiety topic contained the words: feel, worry, stress, study, time, hard) [16], sought to identify words most associated with PTSD and depression status, considered sequences of characters as features, and applied a rule-based approach to build relative counts of N-grams present in PTSD and depression statuses of all users. On the same dataset, Preotiuc-Pietro et al. observed that estimating the age of users adequately identified users who had self-declared a PTSD diagnosis, and that the language predictive of depression and PTSD had large overlap with the language predictive of personality. Character n-gram based methods have been successfully applied to various problems in the text mining domain. Our approach is based on the character n-gram distribution.

3. **Query:** Prognostic modelling using machine learning techniques has been used to predict the risk of kidney graft failure after transplantation. Despite the clinically suitable prediction performance of the models, their decision logic cannot be interpreted by physicians, hindering clinical adoption. eXplainable Artificial

Intelligence (XAI) is an emerging research discipline to investigate methods for explaining machine learning models which are regarded as ‘black-box’ models. In this paper, we present a novel XAI approach to study the influence of time on information gain of donor and recipient factors in kidney graft survival prediction. We trained the most accurate models regardless of their transparency level on subsequent non-overlapping temporal cohorts and extracted faithful decision trees from the models as global surrogate explanations. Comparative exploration of the decision trees reveals insightful information about how the information gain of the input features changes over time. **Query-Focused Summary:** Introduction Over the past decade, there has been an increasing interest in leveraging machine learning (ML) models to aid decision making in critical domains such as healthcare and criminal justice. However, the proprietary nature and increasing complexity of machine learning models poses a severe challenge to understanding these complex black boxes, motivating the need for tools that can explain them in a faithful and interpretable manner. Prior research on interpretable machine learning mainly focused on learning predictive models from scratch which were human understandable. Human interpretability has high importance in a wide range of applications such as medicine and business [4, 8], where results from prediction models are generally presented to a human decision maker/agent who makes the final decision. Interpretable & Explorable Approximations of Black Box Models Himabindu Lakkaraju Stanford University himalv@cs.stanford.edu Ece Kamar Microsoft Research eckamar@microsoft.com Rich Caruana Microsoft Research rcaruana@microsoft.com Jure Leskovec Stanford University jure@cs.stanford.edu ABSTRACT We propose Black Box Explanations through Transparent Approximations (BETA), a novel model agnostic framework for explaining the behavior of any black-box classifier by simultaneously optimizing for fidelity to the original model and interpretability of the explanation. Many approaches have been proposed to directly learn interpretable models (Breiman, 2017; Tibshirani, 1997; Letham et al., 2015; Lakkaraju et al., 2016; Caruana et al., 2015; Kim & Bastani, 2019); however, complex models such as deep neural networks and random forests typically achieve higher accuracy than simpler interpretable models (Ribeiro et al., 2016); thus, it is often desirable to use complex models and then construct post hoc explanations to understand their behavior. As an example, medical diagnosis models [8] may predict a high risk of certain diseases for a patient; a doctor then needs to know the underlying factors to compare with his/her domain knowledge, take the correct action, and communicate with the patient. These experiments show that mimic models can provide insights into black-box models, and demonstrate the advantages of using outcome information. We use this Lending Club example to discuss an insight gained into the black-box model from inspecting feature interactions in the transparent models. To gain insight into the black-box model, we uncover feature regions where the two models are significantly different (Section 2.3), and ask "what could be happening in the black-box model, that could explain the differences we are seeing between the mimic and outcome models?". This allows us to ask, "what could be happening in the black-box model, that could explain the differences we are seeing between the mimic and outcome models?". In addition, similarities between the mimic and outcome models (e.g., on COMPAS in Section 3.2, the Number of Priors feature is modeled very similarly by the two models) increases confidence that the mimic model is a faithful representation of the black-box model, and that any differences observed on other features are meaningful. Because both the mimic and outcome models are trained with the same model class on the same audit data using the same features, the more faithful the mimic model, and the more accurate the outcome model, the more likely it is that observed differences between the mimic and outcome models stem from differences between the black-box model and ground-truth outcomes. A key advantage of using transparent models to audit black-box models is that we do not need to know in advance what to look for. We also carried out user studies in which we asked human subjects to reason about a black box model's behavior using the approximations generated by our approach and other state-of-the-art baselines. Several different kinds of approaches have been proposed to produce interpretable post hoc explanations of black box models. If the black-box model is accurate and generalizes to the audit data, it would predict the ground-truth outcomes in the audit data correctly; the converse is true if the black-box model is not accurate or does not generalize to the audit data. An alternate approach is to provide a global explanation summarizing the black box as a whole (Lakkaraju et al., 2019a; Bastani et al., 2017), typically using an interpretable model.

Chapter 5

MALNIS-DATA: Automatically Building Datasets for Scientific Query-Focused Summarization and Citation Prediction

So far, the tasks of Query-Focused Extractive Summarization and Citation Prediction (QFS/CP) have lagged behind in development when compared to other areas of Scientific Natural Language Processing because of the lack of data [1]. In this work, we propose a methodology to take advantage of existing collections of academic papers to automatically obtain large-scale datasets for these tasks. After applying it to the papers from our research group, we introduce the first large-scale dataset for QFS/CP, composed of 8,695 examples, each one composed of a query, the sentences of the full text from a paper and the relevance labels for each one of them. After testing several classical and state-of-the-art models on this data, we found that these tasks are far from being solved, although they are straight-forward for humans. Surprisingly enough, we found that classical models outperformed modern pre-trained deep language models (sometimes by a large margin), showing that QFS/CP is a fairly unexplored area of Scientific Natural Language Processing. We share our code, data and models for further development of these areas at `https://github.com/jarobyte91/malnis_data`.

5.1 Introduction

Scientists must review and summarize dozens of academic articles frequently to stay up-to-date with the state of the art in their fields. This is especially true before starting a new project, as they need to ensure they incorporate the latest advances in their work. As the number of academic documents keeps increasing yearly, this task has become challenging and time-consuming, especially for students and young researchers [49].

[1] This chapter is an improved version of the paper [70], currently under revision by the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023).

A solution for this problem includes Query-Focused Summarization (QFS) [25] and Citation Prediction (CP) [61] systems, which are helpful to process the extensive collections of papers that practitioners need to analyze. In QFS systems, the objective is to take a long document (or collection of documents) along with the user's query and produce a summary relevant to the query. In CP systems, the objective is to pinpoint the passages where it is appropriate to cite a referenced document. In both cases, the idea is to reduce the amount of text the users need to read and make their task of reviewing literature easier.

Despite their potential applications, creating such systems is not easy [25]. First, it is difficult to determine the *correct* summary or citations from a long document (or document collection), as different people would give a different answer depending on their background and what they are searching for. And second, these tasks usually have small datasets, as having experts read and summarize long documents or extensive collections of documents is a complicated and expensive process.

In this work, we propose a methodology to address the lack of training data for training and evaluating QFS/CP systems by taking advantage of the citations found in peer-reviewed academic publications. The basic idea is that when the authors of a paper cite other documents as references in their work, they implicitly build examples for QFS/CP, as the citing sentences show precisely where the references are relevant. A diagram describing the basic idea behind our approach is shown in Fig. 5.1.

This paper makes the following contributions:

- It proposes a methodology to automatically build datasets for Scientific Query-Focused Extractive Summarization and Citation Prediction directly from raw collections of academic articles. The datasets are composed of three tables: the first contains the text and meta-data of the papers present in the collection, the second one contains the meta-data of the articles cited by the papers in the collection, and the third one contains the citations linking the first two tables. With these tables, it is possible to find examples for these tasks by concatenating the citations to build query-focused summaries or to use them as they are to find citations to predict.

- By applying this methodology to the papers of our reading group, this paper introduces a novel dataset composed of 8,965 examples for the tasks of Scientific

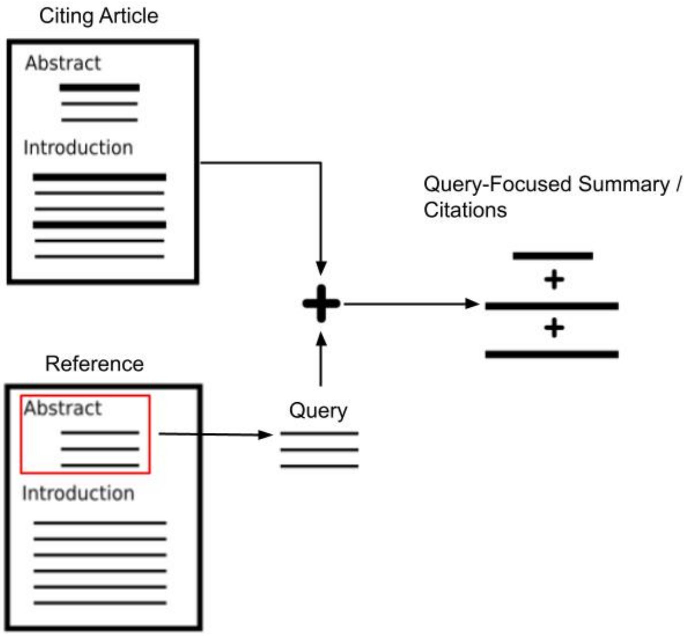

Figure 5.1: Overview of our approach to automatically build datasets for Query-Focused Summarization and Citation Prediction. The basic idea is that when the authors of a paper cite other documents as references in their work, they implicitly build examples for these tasks, as the citing sentences show exactly where the references are relevant. In our approach, the abstract of the referenced article plays the role of the query. In the case of Query-Focused Summarization, the concatenation of the citing sentences makes up the query-focused summary. In the case of Citation Prediction, the citing sentences are the target to predict.

Query-Focused Extractive Summarization and Citation Prediction in the fields of Artificial Intelligence and Natural Language Processing.

- It explores the difficulty of the tasks of Scientific Query-Focused Extractive Summarization and Citation Prediction by applying several classical as well as state-of-the-art methods, showing that, although these tasks are straightforward for humans, even pre-trained deep language models struggle to obtain decent results in them.

The remainder of this paper is organized as follows: Section 2 provides an overview of previous datasets for QFS/CP and the corresponding methodologies employed to build them. Section 3 presents our proposed methodology for leveraging the citations from a document collection to build QFS/CP datasets. Section 4 describes the experiments we performed on the collected data. Section 5 offers a discussion and elaboration on the obtained results. Finally, Section 6 pinpoints our conclusions and directions for future research.

5.2 Related Work

This section discusses previous efforts to build large-scale datasets for QFS/CP. Citation Prediction (primarily studied in the context of Science of Science) is discussed first, while Summarization is discussed second, broken down into (generic) Summarization, Scientific Summarization and Query-Focused Summarization.

5.2.1 Citation Prediction

Within the broader field of citation prediction, a significant portion of research has focused on predicting future citations for existing papers. These studies aim to understand the citation patterns and impact of published works. However, relatively little attention has been given to predicting the citations that a particular paper or publication in progress is likely to make during the writing process. This aspect of citation prediction, which involves anticipating the future referencing behaviour of authors while their paper is still being developed, remains a less explored area of research.

A paper recommendation engine built upon graph-based methods was discussed in [67]. In that work, the authors compare several systems that help scientists improve their academic papers and propose a method that combines several centrality measures to predict the citation graph of a query paper. They evaluated their results on a dataset built from the top 50 most cited articles in the Engineering domain, obtaining promising results.

An agent-based system for identifying citations and ontologies was introduced in [55]. In that work, the authors propose a system that analyzes the user's local collection of academic articles to produce ontologies that help the user find the most related citations by collaborating with other distributed personal citation assistants.

The impact of articles and publications using data-driven methods was discussed in [1]. In that work, the authors analyze how the current bibliographic data can predict important discoveries and identify quantitative patterns that hold across many fields of Science. Nonetheless, they pinpoint the need for transparency in using these techniques, as using them without care could lead to the inhibition of novelty and diversity of Science in general.

5.2.2 Summarization

The field of Summarization has gained significant attention in Natural Language Processing, offering valuable solutions for condensing large volumes of text into concise and coherent summaries. Extractive Summarization techniques involve selecting and presenting important sentences or phrases verbatim from the source document. Another less explored technique is Abstractive Summarization, which attempts to generate summaries by paraphrasing and restructuring the source content.

One of the first methodologies to automatically obtain summaries of news articles was introduced in [35]. This methodology involves querying the news articles obtained from the CNN and DailyMail websites using a variety of combinatorial heuristics to force the models to capture how the different entities in the article relate to each other. They tested the performance of several state-of-the-art methods on their data and demonstrated that their approach is general enough to produce datasets for different domains.

The first large-scale dataset for Multi-Document Summarization was introduced in [2]. This paper exploits the data available at `newser.com`, with 56,216 article-summary pairs, each written by professional editors and with links to the source articles. The novelty of this dataset lies in its size and diversity, surpassing those of previously published datasets. Additionally, they introduced a novel model incorporating Maximal Marginal Relevance into a Pointer-Generator Network, improving the fluency and conciseness of previous multi-document summarization models.

5.2.3 Scientific Summarization

One of the first attempts to create a dataset for scientific document summarization was introduced during the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016) [38]. To build the dataset for the competition, they filtered the most important papers from the ACL Anthology repository (`https://aclanthology.org/`) heuristically. After that, they instructed their annotators to find the citing sentences along with the most important sentences in the citing paper, following the BiomedSumm shared task of the same event.

An enhanced semi-automatic methodology that extends [38] was introduced in [94]. That work incorporates the abstract and incoming citations of a paper to highlight the most important sentences and make a summary out of them. More specifically, they use the sentence relation graph of the paper, the authority scores and the semantic sentence embeddings to estimate the salience of each sentence inside the article with a Graph Convolutional Network [45]. After that, they use a hybrid greedy algorithm to generate the final summary. Finally, they propose a novel algorithm based on Graph Neural Networks that finds the summary spans directly from the scientific papers.

A methodology to automatically obtain summaries from academic articles using presentation and conference talks was introduced in [51]. In that work, they exploit the fact that when a researcher presents a paper, they must express their ideas concretely and concisely, often using key phrases and findings from their research. This means that the talk transcripts or blog posts are often good summaries of the entire article, and hence they introduce a novel unsupervised algorithm based on Hidden Markov Models to align the summaries with the original articles.

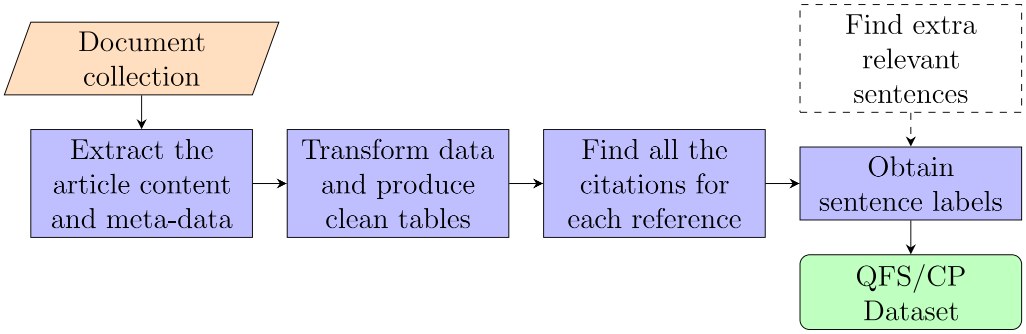
Document collection
Extract the article content and meta-data
Transform data and produce clean tables
Find all the citations for each reference
Find extra relevant sentences
Obtain sentence labels
QFS/CP Dataset

Figure 5.2: Overview of the proposed methodology.

A large-scale dataset composed of 10,148 scientific articles, along with their abstracts, highlighted statements and author-defined keywords, was introduced in [21]. In that work, the authors extracted articles from `http://www.sciencedirect.com/` and proposed a method called HighlightROUGE to extend the dataset automatically. Additionally, they introduced a metric (called AbstractROUGE) to extract summaries by leveraging the abstract of the paper. Finally, they benchmarked several traditional and neural-based summarization methods on their dataset and analyzed how different sections of the paper contributed to the final summary.

5.2.4 Query-Focused Summarization

The first time that the task of Query-Focused Summarization was formally studied was during the 2005 Document Understanding Conference (DUC 2005) [25]. The main purpose of the conference was to study how the variability of the summaries produced by humans affected the performance of the existing methods of the time. To this end, DUC 2005 had a unique summarization task, focusing on the users and their queries instead of the output summaries, as in previous efforts.

In that shared task, the objective was to produce a well-organized and fluent answer to a complex question using a set of 25 to 50 documents. Even while there was a generous allowance of 250 words for each answer, the results revealed that the best systems of the time had a hard time summarizing multiple documents. The two subsequent editions of the conference (DUC 2006 [26] and DUC 2007 [27]) refined the data and results produced in the first conference, and they still are the current reference benchmarks in the field.

Despite their importance and popularity, the DUC datasets lack diversity, as shown by [8]. That paper introduces a new metric called Topic Concentration, which the authors used to show that the DUC datasets already have queries very close to their document collections. Hence, systems designed explicitly for QFS do not significantly improve upon generic summarization methods. Therefore, they introduced TD-QFS, a novel dataset with controlled levels of Topic Concentration, and showed that when evaluated on this data, there is a clear difference between QFS systems and generic summarization systems.

More recently, a novel method and a small dataset for Scientific Query-Focused Summarization was introduced in [85]. For their experiments, they built a new dataset using a two-step approach from the data of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL) [14]. First, they collected additional papers from later publications that reported the results for the same dataset as the submitted papers. Then, they manually selected the most relevant sentences for the queries. When evaluated on this data, their proposed method showed superior results than other methods from the state of the art.

5.3 Methodology

Our methodology is composed of four main steps to extract the content from the papers in a document collection and clean it to obtain the examples that make up the final dataset. It also includes an optional step to improve the quality of the examples found by finding more relevant sentences. An overview of the process is shown in Fig. 5.2.

5.3.1 Extracting the Article Content and Meta-Data

First, all the PDF files from our document collection were processed with Science-Parse [3], an LSTM-based [36] software by AllenAI to extract text from scientific articles. The input for Science-Parse is the raw PDF file of an article, and its output is a JSON file containing the content and meta-data of the paper, such as its title, abstract, sections, information about its authors, the list of its references and the citing sentences from the text, among other fields. An overview of the fields in the JSON file is shown in the top part of Fig. 5.3.

5.3.2 Transforming the Data and Producing Clean Tables

From the set of raw JSON files, three tables are produced: **Papers**, **References** and **Citations**. An overview of the fields in each one is shown in the bottom part of Fig. 5.3.

The **Papers** table contains the information describing each one of the articles in the collection, using the following fields: *paper_id*, *title*, *abstract* and *text*. The *paper_id* fields contain a unique identifier for each paper, obtained after merging and de-duplicating all the papers in the collection. The *title* and *abstract* fields contain the title and abstract of the article obtained after the de-duplication process. Finally, the *text* field contains the full text of the paper, which was obtained as the concatenation of the text present in the *Sections* field of the raw JSON files obtained in the data extraction step.

The **References** table contains the information about the papers cited by the

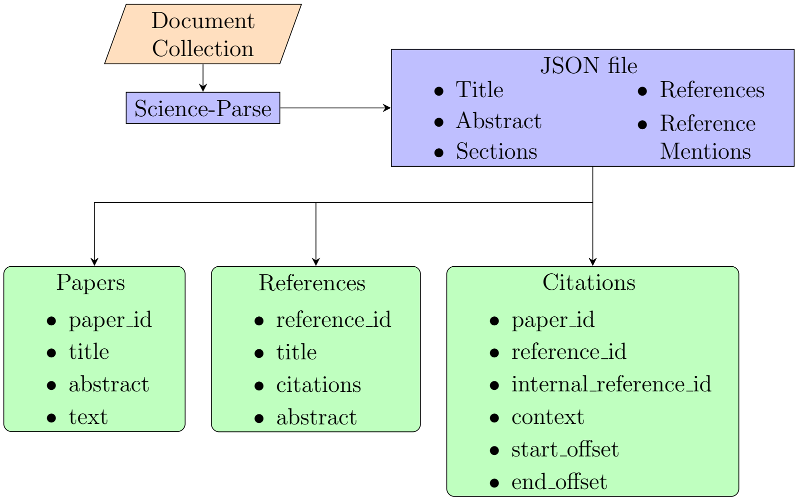

Figure 5.3: The Data Extraction process. First, the content and meta-data of the papers in the collection are extracted using Science-Parse [3] into a collection of JSON files. Then, the JSON files are merged, cleansed and de-duplicated to obtain three clean tables: the *Papers* table contains the information about the papers in the collection, the *References* table contains the information about the references cited by the papers in the collection, and the *Citations* table contains the information about the citations that link the first two tables.

papers from the collection, using the following fields: *reference_id*, *title*, *total_citations* and *abstract*. The *reference_id* field contains a unique identifier for each reference, obtained after merging and de-duplicating the References field of the raw JSON files. The *title* field contains the title of the reference obtained after the de-duplication process. The *total_citations* field contains the total number of times the reference was mentioned in the papers of the collection. The field *abstract* contains the abstract of the reference paper, obtained after crossing the *title* field with the Arxiv dataset [18].

The **Citations** table contains the citations that link the **Papers** and **References** tables, using the following fields: *paper_id*, *reference_id*, *internal_reference_id*, *context*, *start_offset*, *end_offset*. The *paper_id* field contains the unique paper identifier from the **Papers** database. The *reference_id* field contains the unique reference identifier from the **References** database. The *internal_reference_id* field contains the reference number as it appears in the citing paper. The *context* field contains the sentence where the paper cited the reference. The *start_offset* and the *end_offset* fields contain the character span inside the sentence where the reference was cited.

5.3.3 Finding All the Citations for each Reference

Once the clean tables have been produced, it is straightforward to join the **Citations** table with the **Papers** and **References** tables via the unique identifiers of the articles and references to obtain an augmented **Citations** table, which can be grouped by both *paper_id* and *reference_id* to obtain a table in which every row has the following data:

- paper_id
- paper_text
- reference_id
- reference_abstract
- citations_concatenated

5.3.4 Obtaining the Sentence Labels

The final step in our methodology is to produce a True/False label for each one of the sentences from the text of the paper, which encodes its relevance to the query. To do this, the abstract of the reference takes the role of the query, and both the paper text and the concatenated citations have to be tokenized into sentences. Finally, the relevance label for each sentence from the paper is obtained by checking if the sentence is one of the sentences from the concatenated citations. A diagram displaying how the final dataset looks is shown below in Fig. 5.4.

Document (Sequence of Sentences)	$[s_1, s_2, s_3, s_4, s_5, ...]$
Query	q
Sentence Labels	$[l_1, l_2, l_3, l_4, l_5, ...]$

Figure 5.4: Structure of the final dataset. Each example has three elements: a list with the sentences from the full text of the paper, a paragraph query and a list containing the relevance labels for each one of the sentences.

5.3.5 Finding Extra Relevant Sentences

Since each reference was cited by at least one of the papers in the collection, there is guaranteed at least one positive label in each of the examples obtained. However, it is important to note that for many examples, there might be a single positive label in the whole paper. Hence, to obtain more positive labels, we used a greedy approach in which sentences are added one by one to the summary, using ROUGE [54] to compare it to the abstract of the reference. Although this method to find extra relevant sentences is limited and expensive (given how ROUGE works), we found that this augmentation technique worked well in practice. An overview of this process is shown in Fig. 5.5.

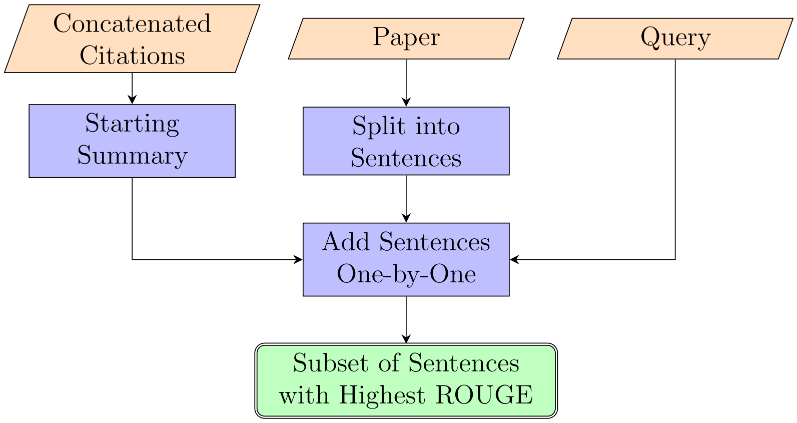

Figure 5.5: The data augmentation process. First, the concatenated citations are taken as the starting summary. Then, the sentence that introduces the best ROUGE score in the current summary when compared against the query is added. This process continues until the ROUGE score stops improving. Ultimately, the selected sentences are a good approximation of the subset of sentences that would give the best ROUGE score.

5.4 Experiments

To measure the effectiveness of the proposed methodology, we applied it to the papers from our reading group and trained a variety of baselines from the current state of the art in NLP. After that, we evaluated the same baselines on data obtained from real users to compare the results and estimate how different is the synthetic data obtained through our methodology from real queries and sentences from document collections.

5.4.1 Data

We applied our methodology to the collection of papers from our reading group, composed of 1,365 PDF files. After grouping the augmented citations table by *reference_id*, we ended up with 10,790 examples with the structure shown in Fig. 5.4. Nonetheless, some examples had documents that were too long to feed into the data augmentation process using our hardware, so after filtering them out, we obtained our final dataset, described in Table 5.1.

Total Size:	8,965 examples
Mean Document Length:	353 sentences
Max Document Length:	4,447 sentences
Mean Fraction of Positive Labels:	3.9%
Train Set Size:	7,172 examples
Development Set Size:	897 examples
Test Set Size:	897 examples

Table 5.1: Details of the final dataset collected after applying our methodology to the papers of our reading group. The original collection consisted of 1,365 PDF files, which produced 10,790 examples. The final dataset was obtained after excluding the examples with documents too long to process with our data augmentation method.

5.4.2 Approach

First, the paper (viewed as a sequence of sentences) and the query are embedded into a Euclidean Space using a representation method or a language model. Then, the query vector is replicated so that each sentence vector is concatenated with a copy of the query vector to produce a sequence of augmented sentence vectors. Next, each component of the sequence of augmented vectors is processed with a binary classifier (which may or may not be aware of the sequence order) to produce a binary label for each sentence, which encodes if the sentence is relevant or not to the query. Finally, the predicted labels are compared with the reference labels using Binary Cross Entropy to train the classifier. A diagram of this process is shown in Fig. 5.6.

5.4.3 Models

To embed the query and the sentences from the papers, we used various classical text representation methods and modern language models. For the classical ones, we used

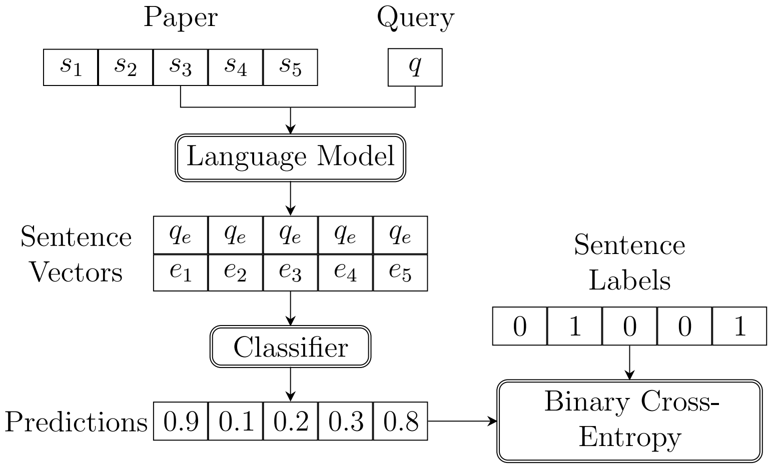

Figure 5.6: Training of the models. First, the sentences from the paper and the query are embedded into a Euclidean Space using a representation method. Then, the query representation is replicated and concatenated with each one of the sentence representations. After that, these augmented sentence vectors are fed into a classifier to estimate the relevance label for each one of them. Finally, the predictions from the classifier are compared with the reference labels via Binary Cross Entropy.

TFIDF [86] based on word uni-grams and character tri-grams. For the modern ones, we used Sentence-BERT [75] and SPECTER [19]. To produce the relevance labels for the sentences, we also used a variety of classical and modern classifiers. For the classical ones, we used the typical Cosine Similarity/Euclidean Distance Classifier and the Multi-Layer Perceptron (MLP). For the modern ones, we used two sequence-aware classifiers, the LSTM [36] and the Transformer [90]. The combinations of language models and classifiers we used are shown in Table 5.2, while the exact hyper-parameters for each one of them can be found in the Appendix.

Classifier	TFIDF Words	TFIDF Chars	Sentence-BERT	SPECTER
Euclidean Distance				X
Cosine Similarity	X	X	X	
Multi-Layer Perceptron (MLP)	X	X	X	X
LSTM			X	X
Transformer			X	X

Table 5.2: Model variations used during the experiments.

5.4.4 Results

Since the objective is to produce a binary label for each sentence, we evaluated the models using both Average Precision and Area under the ROC Curve (ROC AUC), as shown in Table 5.3. Both metrics were computed on each one of the examples in the Test Set using the standard implementation found in [64].

Model	Representation	Average Precision	ROC AUC
Cosine Similarity	TFIDF Words	0.197 ± 0.008	**0.765 ± 0.006**
MLP	TFIDF Chars	0.148 ± 0.006	0.712 ± 0.007
MLP	TFIDF Words	0.145 ± 0.006	0.703 ± 0.007
Cosine Similarity	TFIDF Chars	0.152 ± 0.007	0.701 ± 0.006
LSTM	SPECTER	**0.208 ± 0.018**	0.691 ± 0.009
Transformer	SPECTER	0.193 ± 0.017	0.685 ± 0.010
LSTM	SBERT	0.202 ± 0.018	0.684 ± 0.009
MLP	SPECTER	0.115 ± 0.005	0.678 ± 0.006
MLP	SBERT	0.103 ± 0.005	0.654 ± 0.007
Cosine Similarity	SBERT	0.125 ± 0.006	0.633 ± 0.007
Transformer	SBERT	0.160 ± 0.016	0.628 ± 0.010
Euclidean Distance	SPECTER	0.114 ± 0.006	0.600 ± 0.008

Table 5.3: Mean Average Precision and Mean ROC AUC on the Test Set. The highlighted models are the best ones.

5.5 Discussion

Although some models display decent values of ROC AUC, the Average Precision reveals that the task is challenging for them, as none could obtain more than 0.21 under this metric. Overall, the best models are the Cosine Similarity Classifier on top of TFIDF Word Uni-gram vectors and the LSTM on top of SPECTER embeddings, well above the others. Interestingly, the task appears to be considerably easier when the user is involved in the process, as shown in [73].

Surprisingly, the models based on classical representations (TFIDF Chars and TFIDF Words) performed very well despite their simplicity. Out of these models, it is striking that the Cosine Similarity Classifier on top of TFIDF Words is the best of all the models in terms of ROC AUC. Another interesting fact is that for the Cosine Similarity classifiers, the ones based on TFIDF vectors (character tri-grams and word uni-grams) performed better than the neural-based ones (Sentence-BERT and SPECTER), although TFIDF Chars performed worse than TFIDF Words. As an explanation for these results, it makes sense that looking for matching words between the query and the sentences provides a reasonable baseline for this task.

For the models based on embeddings produced by neural networks, it is interesting to see that the LSTMs performed better than the Transformers. Also, except for the Cosine Similarity/Euclidean Distance Classifier, the SPECTER embeddings appear to be better than the SBERT ones, a trend confirmed with the LSTMs, the Transformers and the MLPs. Finally, it is interesting that the MLPs are on par with the Transformers regarding ROC AUC, although their Average Precision is worse.

To further investigate our results, we computed the fraction of relevant sentences and the mean length of spans of consecutive positive labels for each example in the Train Set, as shown in Table 5.4. This shows that around 4% of the sentences in a given example are relevant and that around 5% of them come in sequences of 2 or more. This explains why the models that are unaware of the sequence order (all but the LSTMs and the Transformers) perform so similarly and why the LSTMs might have an inductive advantage over the Transformers.

Fraction of Relevant Sentences	
Mean	3.90%
STD	2.00%
Min	0.01%
First Quartile	2.43%
Median	3.66%
Third Quartile	5.03%
Max	22.73%

Span Length	Relative Frequency(%)
1	94.953
2	4.739
3	0.269
4	0.032
5	0.005

Table 5.4: Distribution of positive labels in the Train Set.

Furthermore, it is interesting that the Euclidean Distance/Cosine Similarity Classifier based on the SPECTER embeddings is worse than the one based on SBERT embeddings. This is striking, as SPECTER is trained to embed scientific documents. It is important to note that even while it seems that this classifier requires some hyper-parameter tuning, in reality, what matters is the ranking of similarities between the query and the document sentences, which is always based on the pairwise distances of their respective embeddings. Nonetheless, the classifiers based on the SPECTER embeddings outperformed their counterparts based on the SBERT embeddings (sometimes by a large margin), so they appear well-suited for this task.

To finalize the dicussion of our results, we evaluated the models on the ground truth data produced by real users collected using QuOTeS [73], as shown in Table 5.5. Although the results obtained with this dataset are different from the ones obtained during our experiments, it is important to note that this dataset is much smaller (only 23 examples) and that the documents from these examples are much shorter than the ones we obtained with our methodology. Nonetheless, the main conclusions we obtained in our experiments are the same: the classical models still provide strong baselines for the task, the LSTMs outperformed the Transformers and the SPECTER embeddings proved superior than the SBERT ones.

Model	Representation	Average Precision	ROC AUC
MLP	TFIDF Chars	0.664 ± 0.13	$\mathbf{0.682 \pm 0.11}$
MLP	SPECTER	0.652 ± 0.11	0.654 ± 0.11
MLP	TFIDF Words	0.654 ± 0.12	0.650 ± 0.13
Cosine Similarity	TFIDF Chars	$\mathbf{0.674 \pm 0.10}$	0.634 ± 0.13
LSTM	SBERT	0.600 ± 0.11	0.631 ± 0.09
LSTM	SPECTER	0.637 ± 0.10	0.627 ± 0.10
Cosine Similarity	TFIDF Words	0.575 ± 0.11	0.543 ± 0.13
MLP	SBERT	0.600 ± 0.12	0.540 ± 0.13
Euclidean Distance	SPECTER	0.532 ± 0.11	0.505 ± 0.12
Transformer	SBERT	0.545 ± 0.10	0.485 ± 0.12
Transformer	SPECTER	0.556 ± 0.09	0.479 ± 0.12
Cosine Similarity	SBERT	0.526 ± 0.09	0.420 ± 0.12

Table 5.5: Results obtained by the models on the ground truth data collected using QuOTeS [73]. The highlighted models are the best ones.

5.6 Conclusions and Future Work

In this work, we introduced a novel methodology for the automatic creation of datasets for the tasks of Citation Prediction and Scientific Query-Focused Summarization. After applying it to the collection of papers from our reading group, we obtained a dataset composed of 8,965 examples, each with a query, an entire document and the relevance labels for each one of its sentences.

Through several experiments, we have shown that the task of Citation Prediction/Query-Focused Summarization is far from being solved, despite being relatively simple for humans [73]. We have also shown that state-of-the-art systems struggle with this task and that classical, simple models perform better. In particular, the traditional Cosine Similarity Classifier on top of the TFIDF word uni-gram vector outperformed by a large margin the current off-the-shelf methods. Furthermore, we found surprising that, contrary to the current state of the art, a system based on a bidirectional LSTM model outperformed the more complex Transformer. This provides evidence that the task of Scientific Query-Focused Summarization is an interesting challenge inside Scientific Natural Language Processing.

For future work, we would like to investigate why this task is so difficult for the current models. Given the performance shown by Deep Language Models on several benchmarks, it would appear that this task should be easy to solve, but our experiments proved otherwise. Another future direction would be to investigate how the current Generative Deep Language Models like GPT-3 [13] behave on this task and how they can enhance the data collected in this work.

Another important direction for future work is to investigate how to train explainable models using the data produced by the methodology proposed here. Given that once a system for QFS/CP is trained, it is very hard to verify its False Positive Rate, as one would have to trust that the system reviewed correctly the hundreds of sentences present in the paper. One idea in this direction is to filter the section from which the positive examples come from, as usually the *Introduction* and *Related Work* sections contain the most citations.

5.7 Limitations

The first main limitation of the methodology presented in this work is that in some cases, it is difficult to obtain the full query-focused summary or all the citations relevant to a given query. The reason for this is that when the authors of a paper are composing it, they usually stop citing a reference after using it a few times. This means that the citing paper has usually more mentions than the ones found by *Science-Parse*, so sentences that could have been potentially relevant to the query are left out. Unfortunately, we cannot think of a way to fully verify the quality of the data obtained with our method other than reading the full papers and manually extracting all the citations. Nonetheless, a simple solution for this problem is to filter out the examples with very few positive labels. Finally, a more complicated way to overcome this limitation is the optional data augmentation process we included at the end of our methodology.

The second main limitation of this work is that the hardware requirements to use our methodology can be quite high. First, the data augmentation process can be very expensive (as actually happened during our experiments), because if the document is very long, the process of adding *all* the sentences and computing the ROUGE scores of the potential summaries is computationally prohibitive, and it cannot be accelerated with specialized hardware, like GPUs. Second, as outlined in the original repository, *Science-Parse* requires a lot of heap memory, which can be an issue for most users (in our experiments, we ended up using a separate workstation with 32 GB of RAM to extract the raw JSON files). And third, for the examples with very long documents, it is difficult to train the models that are aware of the sequence order (LSTM and Transformer) because of their inherent limitations on the number of sentences they can process at once. Unfortunately, the examples with longer documents are usually the most interesting ones, so future users of the method presented here will have to balance this trade-off between document length and hardware requirements.

5.8 Model Details

In this section, we describe the hyper-parameters needed to implement the models that performed the best in this work. For each one of them, we used Random Search [39] to tune the hyper-parameters on the ranges described below.

5.8.1 Vector Text Representations

Regarding TFIDF representations, we used the standard implementation found in [64] with default parameters for both word uni-grams and character tri-grams. For the neural-based embeddings, we used the standard implementation from `https://www.sbert.net/` [75]. For the general-purpose language model, we used *all-MiniLM-L6-v2*, while for SPECTER, we used *allenai-specter*.

5.8.2 Multi-Layer Perceptron (MLP)

For the MLPs on top of TFIDF representations, we tried from 1 to 4 layers of 100, 200, 300 or 400 hidden units each, trained for 16 epochs. All the other hyper-parameters were left as the default value from the standard implementation found in [64]. For the word uni-grams model, the one that performed the best had a single layer of 400 hidden units, with a total training time of 18.18 hours. For the character tri-grams model, the one that performed the best had three layers of 100 hidden units each, with a total training time of 4.05 hours.

For the MLPs on top of neural-based embeddings, we tried from 1 to 4 layers of 100 to 500 hidden units each, in steps of 50. Each model was trained for 2,000 epochs using Adam [44] with a constant learning rate of 10^{-4} and a L^2 regularization term of 0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} or 10^{-5}. For the Sentence-BERT embeddings, the best model had 3 layers of 300 hidden units each, a regularization value of 10^{-4} and a total training time of 45 minutes. For the SPECTER embeddings, the best model had 4 layers of 450 hidden units each, a regularization value of 0 with a total training time of 91 minutes.

5.8.3 LSTM

For both the models built on top of Sentence-BERT and SPECTER embeddings, we tried from 1 to 4 layers of 100 to 500 hidden units each, in steps of 50. Each model was trained for 2,000 epochs using Adam [44] with a constant learning rate of 10^{-4} and a L^2 regularization term of 0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} or 10^{-5}. For the Sentence-BERT embeddings, the best model had 3 layers of 500 hidden units each, a regularization value of 0 and a total training time of 3.9 hours. For the SPECTER embeddings, the best model had a single layer of 500 hidden units each, a regularization value of 0, with a total training time of 102 minutes.

5.8.4 Transformer

For both the models built on top of Sentence-BERT and SPECTER embeddings, we tried from 2 to 4 Transformer layers, having from 100 to 500 units in its feed-forward networks, in steps of 50, and from 2 to 4 attention heads. Each model was trained for 2,000 epochs using Adam [44] with a constant learning rate of 10^{-4} and a L^2 regularization term of 0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} or 10^{-5}. For the Sentence-BERT embeddings, the best model had 3 layers of 250 units in its feed-forward networks and 4 attention heads, a regularization value of 0 and a total training time of 11.96 hours. For the SPECTER embeddings, the best model had 3 layers, 350 units in its feed-forward networks and 4 attention heads, a regularization value of 10^{-5} with a total training time of 30.25 hours.

5.9 Methodology Output Examples

1. **Query:** Zhang and Yang [149] proposed a lattice-structured LSTM model for Chinese NER, which encodes a sequence of input characters as well as all potential words that match a lexicon. **Query-Focused Summary:** We investigate a lattice-structured LSTM model for Chinese NER, which encodes a sequence of input characters as well as all potential words that match a lexicon. Compared with character-based methods, our model explicitly leverages word and word sequence information. Compared with word-based methods, lattice LSTM does not suffer from segmentation errors. Gated recurrent cells allow our model to choose the most relevant characters and words from a sentence for better NER results. Experiments on various datasets show that lattice LSTM outperforms both word-based and character-based LSTM baselines, achieving the best results.

2. **Query:** Baweja et al. (2018) investigate continual learning of two MRI segmentation tasks with neural networks for countering catastrophic forgetting of the first task when a new one is learned.Baweja et al. (2018) investigate continual learning of two MRI segmentation tasks with neural networks for countering catastrophic forgetting of the first task when a new one is learned. They investigate elastic weight consolidation, a method based on Fisher information to sequentially learn segmentation of normal brain structures and then segmentation of white matter lesions and demonstrate this method reduces catastrophic forgetting, but acknowledge there is a large room for improvement for the challenging setting of continual learning. It is important to quantify

the performance and robustness of a model at every stage of its lifespan. One way to consider stopping could evaluate when the cost of continued training outweighs the cost of errors made by the current model. An existing measure that attempts to quantify the economical value of medical intervention is the Quality-adjusted Life year (QALY), where one QALY equates to one year of healthy life NICE (2013). Could this metric be incorporated into models? At present we cannot quantify the cost of errors made by DL medical imaging applications but doing so could lead to a deeper understanding of how accurate a DL model really ought to be. **Query-Focused Summary:** This work investigates continual learning of two segmentation tasks in brain MRI with neural networks. To explore in this context the capabilities of current methods for countering catastrophic forgetting of the first task when a new one is learned, we investigate elastic weight consolidation, a recently proposed method based on Fisher information, originally evaluated on reinforcement learning of Atari games. We use it to sequentially learn segmentation of normal brain structures and then segmentation of white matter lesions. Our findings show this recent method reduces catastrophic forgetting, while large room for improvement exists in these challenging settings for continual learning.

3. **Query:** It's well known that the key idea lying behind active learning is a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose data from which it learns [13].where , or the class label with the highest posterior probability under the model [13].By contrast, batch-mode active learning allows the learner to query instances in groups, which is better suited to parallel labeling environments or models with slow training procedures [13]. **Query-Focused Summary:** The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose queries, usually in the form of unlabeled data instances to be labeled by an oracle (e.g., a human annotator). Active learning is well-motivated in many modern machine learning problems, where unlabeled data may be abundant or easily obtained, but labels are difficult, time-consuming, or expensive to obtain. This report provides a general introduction

to active learning and a survey of the literature. This includes a discussion of the scenarios in which queries can be formulated, and an overview of the query strategy frameworks proposed in the literature to date. An analysis of the empirical and theoretical evidence for successful active learning, a summary of problem setting variants and practical issues, and a discussion of related topics in machine learning research are also presented.

4. **Query:** While the techniques for neural networks are computationally expensive and approximate, the techniques for mixtures of Gaussians and locally weighted regression are both efficient and accurate [74]. **Query-Focused Summary:** For many types of machine learning algorithms, one can compute the statistically 'optimal' way to select training data. In this paper, we review how optimal data selection techniques have been used with feedforward neural networks. We then show how the same principles may be used to select data for two alternative, statistically-based learning architectures: mixtures of Gaussians and locally weighted regression. While the techniques for neural networks are computationally expensive and approximate, the techniques for mixtures of Gaussians and locally weighted regression are both efficient and accurate. Empirically, we observe that the optimality criterion sharply decreases the number of training examples the learner needs in order to achieve good performance.

5. **Query:** The RACE dataset [17] contains near 100K questions taken from the English exams for middle and high school Chinese students in the age range between 12 to 18, with the answers generated by human experts. **Query-Focused Summary:** We present RACE, a new dataset for benchmark evaluation of methods in the reading comprehension task. Collected from the English exams for middle and high school Chinese students in the age range between 12 to 18, RACE consists of near 28,000 passages and near 100,000 questions generated by human experts (English instructors), and covers a variety of topics which are carefully designed for evaluating the students' ability in understanding and reasoning. In particular, the proportion of questions that requires reasoning is much larger in RACE than that in other benchmark datasets for reading comprehension, and there is a significant gap between the performance of the state-of-the-art models

(43%) and the ceiling human performance (95%). We hope this new dataset can serve as a valuable resource for research and evaluation in machine comprehension. The dataset is freely available at http://www.cs.cmu.edu/ glai1/data/race/ and the code is available at https://github.com/qizhex/RACE_AR_baselines.

Chapter 6

Future Work

The purpose of this section is to point out future interesting research directions to further improve the methods introduced in previous chapters.

An interesting idea to extend the method introduced in Chapter 2 is to improve how it selects its sentences to improve its redundancy. The main reason for this extension is the observation that the method displayed a high Precision but low Recall during the competition, which means that the summaries produced by the method are made of highly relevant sentences, but fail to convey the same information as the reference summaries. Since the summaries are already long enough, this suggests that the sentences included in the summary are very similar to each other.

To help the system include other important sentences, we propose implementing a subsystem that checks if including a candidate sentence would introduce enough novelty and diversity. This can be achieved with a similarity threshold in the simplest of cases, but more sophisticated, graph-based methods can also be used to obtain a sub-graph that is more representative of the general graph structure of the text obtained with the first version of the method.

A first direction to extend the method introduced in Chapter 3 is to apply it to text from sources other than documents processed with systems for OCR, such as Automated Speech Recognition (ASR) or Handwritten Text Recognition (HTR), especially for low-resource languages. What makes this idea interesting is that when looking exclusively at the transcriptions from such systems, some confusing factors (such as speaker, tone and pitch or light, color and style) are no longer present, so a correction method based on characters seems a good option to obtain better generalization.

A second direction is a data augmentation technique based on the addition of random noise (deletion, insertion or replacement of characters) in examples from existing datasets for post-OCR correction. The main observation is that since obtaining data for the correction of documents is particularly expensive, the possibility of creating synthetic examples from real ones is very interesting. However, one important observation is that the noise generation process may be far away from the real one, so care must be taken in order to actually improve the performance of the models.

A first direction to extend the system introduced in 4 is to improve its speed, as shown by the questionnaire responses. Currently, the system is built on Dash, which is a good framework for prototyping, but it is not very efficient and it doesn't scale well in the long run. Hence, a more efficient implementation written directly in JavaScript or Flask could offer better performance. Another alternative would be to find a different hosting for the project, as it is currently in the university's public servers.

A second direction is to load the system with a pre-trained classifier trained for QFS/CP, as currently the system always trains the classifier from scratch. In this way, the system can take advantage of large QFS/CP datasets, such as the one introduced in 5. Another alternative in this direction is to connect it with Large Language Models to obtain better candidate sentences.

An interesting way to extend the methodology introduced in 5 is to improve the quality of the examples it produces. In the simplest case, this can be achieved by filtering out the examples with few positive labels to make the examples richer. In a more complex case, the query and the sentences from the document can be fed into a Large Language Model with advance paraphrasing capabilities to better detect sentences relevant to the query.

An interesting direction to extend the methods presented in Chapters 2, 4 and 5 is to feed the extracted summaries into Generative Language Models like GPT-3 [13] to turn them into abstractive summaries. This would improve their conciseness and cohesiveness, possibly making them better summaries.

To finalize, an overall important future direction for the work presented in this thesis is to find more settings where *local methods* are useful: as shown in this thesis, they can be valuable in settings where resources are limited, as they allow the application of more heavy, state-of-the-art techniques where normally it would not be possible.

Some examples of tasks where *local methods* would be useful are the following:

General-Purpose Spell Checking, Part-Of-Speech (POS) Tagging and Correction. Also, it would be interesting to explore how well they can be applied to tasks with somewhat local dependencies, like Machine Translation and Reading Comprehension.

Chapter 7

Conclusions

All the methods introduced in this thesis exemplify the same core idea: although some problems involving document-level NLP seem to require the treatment of long-range dependencies, they can be solved effectively with local methods.

As shown in [71], the production of long summaries from full scientific documents can be reasonably solved by splitting the input document into sentences and cleverly selecting the most important sentences to include in the summary. In [72], we showed that the detection and correction of anomalies at the character level in historical documents processed with OCR engines can be reasonably solved by splitting the long character string into n-grams, correct each one of them independently and merging them together in a coherent way. In [73], we showed that the task of Query-Focused Summarization of Scientific Documents can be reasonably solved via splitting the input documents into sentences and applying Active Learning techniques on them, involving the user in the process. Finally, in [70], we showed that assuming having enough training examples, the tasks of Query-Focused Summarization of Scientific Documents and Citation Prediction can be solved better with classical models than with more modern, complicated ones.

The common theme is that the dependencies to solve these problems can be long, but can be bounded inside a sufficiently large window of constant width around every character, token or sentence in the text. This dramatically reduces the amount of training examples required to successfully train neural networks for these problems, which in turn cuts down the hardware requirements needed to apply them.

This observation is what makes local methods feasible, since trying to apply local methods to solve problems with long-range dependencies like Machine Translation, Abstractive Summarization or Machine Comprehension would limit the model's capacity to learn from the training examples.

Overall, we observed that local methods require very modest hardware to be trained, as all the methods presented in this thesis were trained using a single GPU, as opposed to the current state-of-the-art methods, which usually require from 4 to 8 high-end GPUs to be trained. However, the training from the models can sometimes be long, for example the Czech model in [72] or the Transformer model on the SPECTER embeddings in [70].

Finally, we observed that local methods can be quite sample-efficient, requiring relatively few examples to be trained successfully. For instance, the models for seven out of the nine languages trained in [72] (Bulgarian, Czech, English, Spanish, Dutch, Polish and Slovak) were trained using slightly less than 150 examples, which is a tiny amount of data compared to the current state-of-the-art methods, which usually require hundreds of thousands of examples to be trained successfully.

Appendix A

Copyright Agreements

Part A. Information for Proceedings

Title of Submission: Unsupervised document summarization using pre-trained sentence embeddings and graph centrality

Authors: Juan Ramirez and Evangelos Milios

Abstract: Review your abstract in the box below, and make any changes you deem to be appropriate.

This paper describes our submission for the LongSumm task in SDP 2021. We propose a method for incorporating sentence embeddings produced by deep language models into extractive summarization techniques based on graph centrality in an unsupervised manner.The proposed method is simple, fast, can summarize any kind of document of any size and can satisfy any length constraints for the summaries produced. The method offers competitive performance to more sophisticated supervised methods and can serve as a proxy for abstractive summarization techniques

Preview

Part B. Copyright

Please read the following copyright form. Then type your name below, which signifies your agreement to the terms herein. Authors of exceptional cases should type the string "NA" to the "signature" field and forward an authorized "License to Publish" or equivalent forms to the PC chairs.

Copyright to the above work (including, without limitation, the right to publish the work in whole or in part in any and all forms and media, now or hereafter known) is hereby transferred to the Association for Computational Linguistics (ACL), effective as of the date of this agreement, on the understanding that the work has been accepted for presentation at a meeting sponsored by the ACL and for publication in the proceedings of that meeting. However, each of the authors and the employers for whom the work was performed reserve all other rights, specifically including the following: (1) All proprietary rights other than copyright and publication rights transferred to ACL; (2) The right to publish in a journal or collection or to be used in future works of the author's own (such as articles or books) all or part of this work, provided that acknowledgment is given to the ACL and a full citation to its publication in the particular proceedings is included; (3) The right to make oral presentation of the material in any forum; (4) The right to make copies of the work for internal distribution within the author's organization and for external distribution as a preprint, reprint, technical report, or related class of document.

Exceptions: Some of the foregoing conditions will not apply if the work is not copyrighted or has a non-transferrable copyright, because it has been produced by government employees acting within the scope of their employment. The exceptions are:

- The work has been produced by government employees acting within the scope of their employment and is not copyrighted.
- The work has been produced by government employees acting within the scope of their employment and has non-transferrable copyright.
- The work has been produced by government employees acting within the scope of their employment and the government reserves the right to reproduce the work for government purposes.

I also hereby grant permission to ACL to record my presentation on video and disseminate it under a Creative Commons attribution license.

By typing my signature below, I confirm that all authors of the work have agreed to the above and that I am authorized to execute this agreement on their behalf.

Signature (full name)

(*) Juan Antonio Ramirez-Orta

Job title (if not author):

PhD student

Name of organization

(*) Dalhousie University

Physical address of organization (*)

6385 South Street, Halifax, CA B3H 4R2

Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303 USA

AAAI COPYRIGHT FORM

Title of Article/Paper: Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models

Publication in Which Article/Paper Is to Appear: Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-22)

Author's Name(s): Juan Antonio Ramirez-Orta

Please type or print your name(s) as you wish it (them) to appear in print

PART A – COPYRIGHT TRANSFER FORM

The undersigned, desiring to publish the above article/paper in a publication of the Association for the Advancement of Artificial Intelligence, (AAAI), hereby transfer their copyrights in the above article/paper to the Association for the Advancement of Artificial Intelligence (AAAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the article/paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights current exist or hereafter come into effect, and also the exclusive right to create electronic versions of the article/paper, to the extent that such right is not subsumed under copyright.

The undersigned warrants that they are the sole author and owner of the copyright in the above article/paper, except for those portions shown to be in quotations; that the article/paper is original throughout; and that the undersigned right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify AAAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense AAAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to AAAI in the article/paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorneys' fees incurred therein.

Returned Rights

In return for these rights, AAAI hereby grants to the above author(s), and the employer(s) for whom the work was performed, royalty-free permission to:

1. Retain all proprietary rights other than copyright (such as patent rights).
2. Personal reuse of all or portions of the above article/paper in other works of their own authorship. This does not include granting third-party requests for reprinting, republishing, or other types of reuse. AAAI must handle all such third-party requests.
3. Reproduce, or have reproduced, the above article/paper for the author's personal use, or for company use provided that AAAI copyright and the source are indicated, and that the copies are not used in a way that implies AAAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the article/paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own web page or ftp site. Such web page or ftp site, in addition to the aforementioned requirements of this Paragraph, shall not post other AAAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without AAAI's written permission.
4. Make limited distribution of all or portions of the above article/paper prior to publication.
5. In the case of work performed under a U.S. Government contract or grant, AAAI recognized that the U.S. Government has royalty-free permission to reproduce all or portions of the above Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract or grant so requires.

In the event the above article/paper is not accepted and published by AAAI, or is withdrawn by the author(s) before acceptance by AAAI, this agreement becomes null and void.

(1)

[redacted]

Author/Authorized Agent for Joint Author's Signature

21-04-2022

Date

Dalhousie University

Employer for whom work was performed

Title (if not author)

(For jointly authored Works, all joint authors should sign unless one of the authors has been duly authorized to act as agent for the others.)

PART B – U.S. GOVERNMENT EMPLOYEE CERTIFICATION

This will certify that all authors of the above article/paper are employees of the U.S. Government and performed this work as part of their employment, and that the article/paper is therefore not subject to U.S. copyright protection. The undersigned warrants that they are the sole author/translator of the above article/paper, and that the article/paper is original throughout, except for those portions shown to be in quotations.

(2)

U.S. Government Employee Authorized Signature

Date

Name of Government Organization

Title (if not author)

*(Please read and sign and return Part B **only** if you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign only at signature line (1) in Part A and see item 5 under returned rights. Authors who are U.S. government employees should also sign signature line (1) in Part A above to enable AAAI to claim and protect its copyright in international jurisdictions.)*

PART C–CROWN COPYRIGHT CERTIFICATION

This will certify that all authors of the above article/paper are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties , and that the article/paper is therefore subject to Crown Copyright and is not assigned to AAAI as set forth in the first sentence of the Copyright Transfer Section in Part A. The undersigned warrants that they are the sole author/translator of the above article/paper, and that the article/paper is original throughout, except for those portions shown to be in quotations, and acknowledges that AAAI has the right to publish, distribute, and reprint the Work in all forms and all media.

(3)

British or British Commonwealth Government Employee Authorized Signature

Date

Name of Government Organization

Title (if not author)

*(Please read and sign and return Part C **only** if you are a British or British Commonwealth Government employee and the Work is subject to Crown Copyright. Authors who are British or British Commonwealth government employees should also sign signature line (1) in Part A above to indicate their acceptance of all terms other than the copyright transfer.)*

Licence to Publish Proceedings Papers

SPRINGER NATURE

Licensee	Springer Nature Switzerland AG	(the 'Licensee')
Title of the Proceedings Volume/Edited Book or Conference Name:	The 17th International Conference on Document Analysis and Recognition (ICDAR 2023)	(the 'Volume')
Volume Editor(s) Name(s):	Gernot A. Fink, Rajiv Jain, Koichi Kise, Richard Zanibbi	
Proposed Title of the Contribution:	QuOTeS: Query-Oriented Technical Summarization	(the 'Contribution')
Series: The Contribution may be published in the following series	A Springer Nature Computer Science book series (CCIS, LNAI, LNBI, LNBIP or LNCS)	
Author(s) Full Name(s):	Juan Ramirez-Orta, Eduardo Xamena, Ana Maguitman, Axel J. Soto, Flavia P. Zanoto and Evangelos Milios	(the 'Author')

When Author is more than one person the expression "Author" as used in this Agreement will apply collectively unless otherwise indicated.

Corresponding Author Name:	Juan Ramirez-Orta	
Instructions for Authors	https://resource-cms.springernature.com/springer-cms/rest/v1/content/19242230/data/	(the 'Instructions for Authors')

1 Grant of Rights

a) For good and valuable consideration, the Author hereby grants to the Licensee the perpetual, exclusive, world-wide, assignable, sublicensable and unlimited right to: publish, reproduce, copy, distribute, communicate, display publicly, sell, rent and/or otherwise make available the contribution identified above, including any supplementary information and graphic elements therein (e.g. illustrations, charts, moving images) (the ‘Contribution’) in any language, in any versions or editions in any and all forms and/or media of expression (including without limitation in connection with any and all end-user devices), whether now known or developed in the future. Without limitation, the above grant includes: (i) the right to edit, alter, adapt, adjust and prepare derivative works; (ii) all advertising and marketing rights including without limitation in relation to social media; (iii) rights for any training, educational and/or instructional purposes; (iv) the right to add and/or remove links or combinations with other media/works; and (v) the right to create, use and/or license and/or sublicense content data or metadata of any kind in relation to the Contribution (including abstracts and summaries) without restriction. The above rights are granted in relation to the Contribution as a whole or any part and with or in relation to any other works.

b) Without limiting the rights granted above, Licensee is granted the rights to use the Contribution for the purposes of analysis, testing, and development of publishing- and research-related workflows, systems, products, projects, and services; to confidentially share the Contribution with select third parties to do the same; and to retain and store the Contribution and any associated correspondence/files/forms to maintain the historical record, and to facilitate research integrity investigations. The grant of rights set forth in this clause (b) is irrevocable.

c) If the Licensee elects not to publish the Contribution for any reason, all publishing rights under this Agreement as set forth in clause 1a above will revert to the Author.

2 Copyright

Ownership of copyright in the Contribution will be vested in the name of the Author. When reproducing the Contribution or extracts from it, the Author will acknowledge and reference first publication in the Volume.

3 Use of Contribution Versions

a) For purposes of this Agreement: (i) references to the “Contribution” include all versions of the Contribution; (ii) “Submitted Manuscript” means the version of the Contribution as first submitted by the Author prior to peer review; (iii) “Accepted Manuscript” means the version of the Contribution accepted for publication, but prior to copy-editing and typesetting; and (iv) “Version of Record” means the version of the Contribution published by the Licensee, after copy-editing and typesetting. Rights to all versions of the Manuscript are granted on an exclusive basis, except for the Submitted Manuscript, to which rights are granted on a non-exclusive basis.

b) The Author may make the Submitted Manuscript available at any time and under any terms (including, but not limited to, under a CC BY licence), at the Author’s discretion. Once the Contribution has been published, the Author will include an acknowledgement and provide a link to the Version of Record on the publisher’s website: “This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in [insert volume title], and is available online at https://doi.org/[insert DOI]”.

c) The Licensee grants to the Author (i) the right to make the Accepted Manuscript available on their own personal, self-maintained website immediately on acceptance, (ii) the right to make the Accepted Manuscript available for public release on any of the following twelve (12) months after first publication (the "Embargo Period"): their employer’s internal website; their institutional and/or funder repositories. Accepted Manuscripts may be deposited in such repositories immediately upon acceptance, provided they are not made publicly available until after the Embargo Period.
The rights granted to the Author with respect to the Accepted Manuscript are subject to the conditions that (i) the Accepted Manuscript is not enhanced or substantially reformatted by the Author or any third party, and (ii) the Author includes on the Accepted Manuscript an acknowledgement in the following form, together with a link to the published version on the publisher’s website: “This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/[insert DOI]. Use of this Accepted Version is subject to the publisher’s Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms”. Under no circumstances may an Accepted Manuscript be shared or distributed under a Creative Commons or other form of open access licence.
Any use of the Accepted Manuscript not expressly permitted under this subclause (c) is subject to the Licensee’s prior consent.

d) The Licensee grants to Author the following non-exclusive rights to the Version of Record, provided that, when reproducing the Version of Record or extracts from it, the Author acknowledges and references first publication in the Volume according to current citation standards. As a minimum, the acknowledgement must state: “First published in [Volume, page number, year] by Springer Nature”.

i. to reuse graphic elements created by the Author and contained in the Contribution, in presentations and other works created by them;

ii. the Author and any academic institution where they work at the time may reproduce the Contribution for the purpose of course teaching (but not for inclusion in course pack material for onward sale by libraries and institutions);

iii. to reuse the Version of Record or any part in a thesis written by the same Author, and to make a copy of that thesis available in a repository of the Author(s)' awarding academic institution, or other repository required by the awarding academic institution. An acknowledgement should be included in the citation: "Reproduced with permission from Springer Nature";

iv. to reproduce, or to allow a third party to reproduce the Contribution, in whole or in part, in any other type of work (other than thesis) written by the Author for distribution by a publisher after an embargo period of 12 months; and

v. to publish an expanded version of their Contribution provided the expanded version (i) includes at least 30% new material (ii) includes an express statement specifying the incremental change in the expanded version (e.g., new results, better description of materials, etc.).

4 Warranties & Representations

Author warrants and represents that:

a)

i. the Author is the sole copyright owner or has been authorised by any additional copyright owner(s) to grant the rights defined in clause 1,

ii. the Contribution does not infringe any intellectual property rights (including without limitation copyright, database rights or trade mark rights) or other third party rights and no licence from or payments to a third party are required to publish the Contribution,

iii. the Contribution has not been previously published or licensed, nor has the Author committed to licensing any version of the Contribution under a licence inconsistent with the terms of this Agreement,

iv. if the Contribution contains materials from other sources (e.g. illustrations, tables, text quotations), Author has obtained written permissions to the extent necessary from the copyright holder(s), to license to the Licensee the same rights as set out in clause 1 but on a non-exclusive basis and without the right to use any graphic elements on a stand-alone basis and has cited any such materials correctly;

b) all of the facts contained in the Contribution are according to the current body of research true and accurate;

c) nothing in the Contribution is obscene, defamatory, violates any right of privacy or publicity, infringes any other human, personal or other rights of any person or entity or is otherwise unlawful and that informed consent to publish has been obtained for any research participants;

d) nothing in the Contribution infringes any duty of confidentiality owed to any third party or violates any contract, express or implied, of the Author;

e) all institutional, governmental, and/or other approvals which may be required in connection with the research reflected in the Contribution have been obtained and continue in effect;

f) all statements and declarations made by the Author in connection with the Contribution are true and correct;

g) the signatory who has signed this Agreement has full right, power and authority to enter into this Agreement on behalf of all of the Authors; and

h) the Author complies in full with: i. all instructions and policies in the Instructions for Authors, ii. the Licensee's ethics rules (available at https://www.springernature.com/gp/authors/book-authors-code-of-conduct), as may be updated by the Licensee at any time in its sole discretion.

5 Cooperation

a) The Author will cooperate fully with the Licensee in relation to any legal action that might arise from the publication of the Contribution, and the Author will give the Licensee access at reasonable times to any relevant accounts, documents and records within the power or control of the Author. The Author agrees that any Licensee affiliate through which the Licensee exercises any rights or performs any obligations under this Agreement is intended to have the benefit of and will have the right to enforce the terms of this Agreement.

b) Author authorises the Licensee to take such steps as it considers necessary at its own expense in the Author's name(s) and on their behalf if the Licensee believes that a third party is infringing or is likely to infringe copyright in the Contribution including but not limited to initiating legal proceedings.

6 Author List

Changes of authorship, including, but not limited to, changes in the corresponding author or the sequence of authors, are not permitted after acceptance of a manuscript.

7 Post Publication Actions

The Author agrees that the Licensee may remove or retract the Contribution or publish a correction or other notice in relation to the Contribution if the Licensee determines that such actions are appropriate from an editorial, research integrity, or legal perspective.

8 Controlling Terms

The terms of this Agreement will supersede any other terms that the Author or any third party may assert apply to any version of the Contribution.

9 Governing Law

This Agreement shall be governed by, and shall be construed in accordance with, the laws of Switzerland. The courts of Zug, Switzerland shall have the exclusive jurisdiction.

Signed for and on behalf of the Author	Print Name:	Date:
[redacted]	Juan Antonio Ramirez-Orta	April 27, 2023

Address:	1061 Wellington Street, Apt 202, Halifax, NS, Canada, B3H3A1
Email:	mat.juan.ramirez@gmail.com

Bibliography

[1] Aaron Clauset and Daniel B. Larremore and Roberta Sinatra. Data-driven predictions in the science of science. *Science*, 355(6324):477–480, 2017.

[2] Alexander R. Fabbri and Irene Li and Tianwei She and Suyi Li and Dragomir R. Radev. Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. *arXiv preprint arXiv:1906.01749*, 2019.

[3] AllenAI. Science Parse. GitHub Repository, `https://github.com/allenai/science-parse`, October 2019. visited on April 23, 2021.

[4] Amrhein, Chantal and Clematide, Simon. Supervised ocr error detection and correction using statistical and neural machine translation methods. *Journal for Language Technology and Computational Linguistics (JLCL)*, 33(1):49–76, 2018.

[5] Bajaj, Payal and Campos, Daniel and Craswell, Nick and Deng, Li and Gao, Jianfeng and Liu, Xiaodong and Majumder, Rangan and McNamara, Andrew and Mitra, Bhaskar and Nguyen, Tri and Rosenberg, Mir and Song, Xia and Stoica, Alina and Tiwary, Saurabh and Wang, Tong. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[6] Banerjee, Satanjeev and Lavie, Alon. 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.

[7] Bassil, Youssef and Alwani, Mohammad. OCR Post-Processing Error Correction Algorithm Using Google's Online Spelling Suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3(1), 2012.

[8] Baumel, Tal and Cohen, Raphael and Elhadad, Michael. Topic Concentration in Query Focused Summarization Datasets. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[9] Bayatmakou, Farnoush and Mohebi, Azadeh and Ahmadi, Abbas. An interactive query-based approach for summarizing scientific documents. *Information Discovery and Delivery*, 50(2):176–191, 2021.

[10] Beltagy, Iz and Cohan, Arman and Feigenblat, Guy and Freitag, Dayne and Ghosal, Tirthankar and Hall, Keith and Herrmannova, Drahomira and Knoth, Petr and Lo, Kyle and Mayr, Philipp and Patton, Robert M. and Shmueli-Scheuer, Michal and de Waard, Anita and Wang, Kuansan and Wang, Lucy Lu, editor. *Proceedings of the Second Workshop on Scholarly Document Processing*, Online, June 2021. Association for Computational Linguistics.

[11] Bengio, Yoshua and Ducharme, Réjean and Vincent, Pascal and Janvin, Christian. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3(null), 2003.

[12] Brooke, John. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[13] Brown, Tom B. and Mann, Benjamin and Ryder, Nick and Subbiah, Melanie and Kaplan, Jared and Dhariwal, Prafulla and Neelakantan, Arvind and Shyam, Pranav and Sastry, Girish and Askell, Amanda and Agarwal, Sandhini and Herbert-Voss, Ariel and Krueger, Gretchen and Henighan, Tom and Child, Rewon and Ramesh, Aditya and Ziegler, Daniel M. and Wu, Jeffrey and Winter, Clemens and Hesse, Christopher and Chen, Mark and Sigler, Eric and Litwin, Mateusz and Gray, Scott and Chess, Benjamin and Clark, Jack and Berner, Christopher and McCandlish, Sam and Radford, Alec and Sutskever, Ilya and Amodei, Dario. Language Models Are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[14] Cabanac, Guillaume and Chandrasekaran, Muthu Kumar and Frommholz, Ingo and Jaidka, Kokil and Kan, Min-Yen and Mayr, Philipp and Wolfram, Dietmar, editor. *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, June 2016.

[15] Chiron, Guillaume and Doucet, Antoine and Coustaty, Mickaël and Moreux, Jean-Philippe. ICDAR 2017 competition on post-OCR text correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1423–1428. IEEE, 2017.

[16] Cho, Kyunghyun and van Merriënboer, Bart and Gulcehre, Caglar and Bahdanau, Dzmitry and Bougares, Fethi and Schwenk, Holger and Bengio, Yoshua. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[17] Cho, Kyunghyun and van Merriënboer, Bart and Gulcehre, Caglar and Bahdanau, Dzmitry and Bougares, Fethi and Schwenk, Holger and Bengio, Yoshua. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[18] Clement, Colin B and Bierbaum, Matthew and O'Keeffe, Kevin P and Alemi, Alexander A. On the Use of ArXiv as a Dataset. *arXiv preprint arXiv:1905.00075*, 2019.

[19] Cohan, Arman and Feldman, Sergey and Beltagy, Iz and Downey, Doug and Weld, Daniel. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online, July 2020. Association for Computational Linguistics.

[20] Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[21] Collins, Ed and Augenstein, Isabelle and Riedel, Sebastian. A Supervised Approach to Extractive Summarisation of Scientific Papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[22] Cormack, Gordon V. and Grossman, Maura R. 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.

[23] Cormack, Gordon V. and Grossman, Maura R. 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.

[24] Cormack, Gordon V. and Grossman, Maura R. 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.

[25] Dang, Hoa Trang. Overview of DUC 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[26] Dang, Hoa Trang. Overview of DUC 2006. In *Proceedings of the document understanding conference*, volume 2006, pages 1–10, 2006.

[27] Dang, Hoa Trang. Overview of DUC 2007. In *Proceedings of the document understanding conference*, volume 2007, pages 1–53, 2007.

[28] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[29] Erera, Shai and Shmueli-Scheuer, Michal and Feigenblat, Guy and Peled Nakash, Ora and Boni, Odellia and Roitman, Haggai and Cohen, Doron and Weiner, Bar and Mass, Yosi and Rivlin, Or and Lev, Guy and Jerbi, Achiya and Herzig, Jonathan and Hou, Yufang and Jochim, Charles and Gleize, Martin and Bonin, Francesca and Bonin, Francesca and Konopnicki, David. 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.

[30] Erkan, Günes and Radev, Dragomir R. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004.

[31] Ester, Martin and Kriegel, Hans-Peter and Sander, Jörg and Xu, Xiaowei. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[32] Gerard Salton and Amit Singhal and Mandar Mitra and Chris Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207, 1997. Methods and Tools for the Automatic Construction of Hypertext.

[33] Gerard Salton and James Allan and Chris Buckley and Amit Singhal. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science*, 264(5164):1421–1426, 1994.

[34] H. P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[35] Hermann, Karl Moritz and Kočiský Tomášand Grefenstette, Edward and Espeholt, Lasse and Kay, Will and Suleyman, Mustafa and Blunsom, Phil. Teaching Machines to Read and Comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[36] Hochreiter, Sepp and Schmidhuber, Jürgen. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[37] Iz Beltagy and Matthew E. Peters and Arman Cohan. Longformer: The Long-Document Transformer. *arXiv*, cs.CL/2004.05150, 2020.

[38] Jaidka, Kokil and Yasunaga, Michihiro and Chandrasekaran, Muthu Kumar and Radev, Dragomir and Kan, Min-Yen. The CL-SciSumm Shared Task 2018: Results and Key Insights. *arXiv preprint arXiv:1909.00764*, 2019.

[39] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[40] Jeffrey Pennington and Richard Socher and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[41] Jiacheng Xu and Zhe Gan and Yu Cheng and Jingjing Liu. Discourse-Aware Neural Extractive Model for Text Summarization. *CoRR*, abs/1910.14142, 2019.

[42] Jingqing Zhang and Yao Zhao and Mohammad Saleh and Peter J. Liu. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv preprint arXiv:1912.08777*, 2019.

[43] Ju, Jiaxin and Liu, Ming and Gao, Longxiang and Pan, Shirui. Monash-Summ@LongSumm 20 SciSummPip: An Unsupervised Scientific Paper Summarization Pipeline. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 318–327, Online, November 2020. Association for Computational Linguistics.

[44] Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] Kipf, Thomas N and Welling, Max. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[46] Kolak, Okan and Resnik, Philip. OCR Error Correction Using a Noisy Channel Model. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 257–262, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[47] Kukich, Karen. Techniques for Automatically Correcting Words in Text. *ACM Comput. Surv.*, 24(4):377–439, December 1992.

[48] Kullback, Solomon and Leibler, Richard A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[49] Landhuis, Esther. Scientific literature: Information overload. *Nature*, 535(7612):457–458, 2016.

[50] Leuski, Anton and Lin, Chin-Yew and Hovy, Eduard. 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.

[51] Lev, Guy and Shmueli-Scheuer, Michal and Herzig, Jonathan and Jerbi, Achiya and Konopnicki, David. TalkSumm: A Dataset and Scalable Annotation Method for Scientific Paper Summarization Based on Conference Talks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2125–2131, Florence, Italy, July 2019. Association for Computational Linguistics.

[52] Li, Lei and Xie, Yang and Liu, Wei and Liu, Yinan and Jiang, Yafei and Qi, Siya and Li, Xingyuan. CIST@CL-SciSumm 2020, LongSumm 2020: Automatic Scientific Document Summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234, Online, November 2020. Association for Computational Linguistics.

[53] Likert, Rensis. A technique for the measurement of attitudes. *Archives of psychology*, 140:1–55, 1932.

[54] Lin, Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[55] Ma, C. and Feng, J.Q. and Yang, Z. and Wu, Q.H. Agent-based personal article citation assistant. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 702–705, 2005.

[56] Mihalcea, Rada and Tarau, Paul. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[57] Mihalcea, Rada and Tarau, Paul. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[58] Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges and L. Bottou and M. Welling and Z. Ghahramani and K. Q. Weinberger, editor, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[59] Nguyen, Thi-Tuyet-Hai and Jatowt, Adam and Coustaty, Mickael and Nguyen, Nhu-Van and Doucet, Antoine. Post-OCR Error Detection by Generating Plausible Candidates. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 876–881, 2019.

[60] Nguyen, Thi Tuyet Hai and Jatowt, Adam and Nguyen, Nhu-Van and Coustaty, Mickael and Doucet, Antoine. Neural Machine Translation with BERT for Post-OCR Error Detection and Correction. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 333–336, 2020.

[61] P, Radhika Dileep and R, Deepthi L. Link Prediction in Citation Networks: A Survey. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pages 1194–1200, 2022.

[62] Pallets Projects. Flask: web development, one drop at a time. Python package, `https://flask.palletsprojects.com`, 2010. Visited on August 30, 2022.

[63] Papineni, Kishore and Roukos, Salim and Ward, Todd and Zhu, Wei-Jing. 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.

[64] Pedregosa, Fabian and Varoquaux, Gaël and Gramfort, Alexandre and Michel, Vincent and Thirion, Bertrand and Grisel, Olivier and Blondel, Mathieu and Prettenhofer, Peter and Weiss, Ron and Dubourg, Vincent and Vanderplas, Jake and Passos, Alexandre and Cournapeau, David and Brucher, Matthieu and Perrot, Matthieu and Duchesnay, Édouard. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, nov 2011.

[65] Philip Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994.

[66] Plotly. Dash. Python package, `https://plotly.com/dash`, 2013. Visited on August 30, 2022.

[67] Porshnev, Alexander and Kazakov, Maxim. Possible Ways of Applying Citations Network Analysis to a Scientific Writing Assistant. In Batsyn, Mikhail V. and Kalyagin, Valery A. and Pardalos, Panos M., editor, *Models, Algorithms and Technologies for Network Analysis*, pages 119–126, Cham, 2014. Springer International Publishing.

[68] Radford, Alec and Narasimhan, Karthik and Salimans, Tim and Sutskever, Ilya and others. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[69] Ramesh, Animesh and Srinivasa, K. and .N, Pramod. SentenceRank — A graph based approach to summarize text. In *5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014*, pages 177–182, 02 2014.

[70] Ramirez-Orta, Juan and Maguitman, Ana and Soto, Axel J. and Milios, Evangelos. MALNIS-DATA: Automatically Building Datasets for Scientific Query-Focused Summarization and Citation Prediction. Online preprint, `https://web.cs.dal.ca/~juanr/resources/downloads/malnis_data.pdf`, 2023. Currently under revision at EMNLP 2023. Visited on July 15, 2023.

[71] Ramirez-Orta, Juan and Milios, Evangelos. Unsupervised document summarization using pre-trained sentence embeddings and graph centrality. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 110–115, Online, June 2021. Association for Computational Linguistics.

[72] Ramirez-Orta, Juan and Xamena, Eduardo and Maguitman, Ana and Milios, Evangelos and Soto, Axel J. Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11192–11199, June 2022.

[73] Ramirez-Orta, Juan and Xamena, Eduardo and Maguitman, Ana and Soto, Axel J. and Zanoto, Flavia P. and Milios, Evangelos. QuOTeS: Query-Oriented Technical Summarization. *arXiv preprint arXiv:2306.11832*, June 2023.

[74] Ray W. Smith. The Extraction and Recognition of Text from Multimedia Document Images. *PhD Thesis, University of Bristol*, 1987.

[75] Reimers, Nils and Gurevych, Iryna. 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.

[76] Rewon Child and Scott Gray and Alec Radford and Ilya Sutskever. Generating Long Sequences with Sparse Transformers. *arXiv*, cs.LG/1904.10509, 2019.

[77] Rice, Stephen V. and Jenkins Frank R. and Nartker, Thomas A. The Fourth Annual Test of OCR Accuracy. In *Technical Report 95-03*, Las Vegas, 1995. Information Science Research Institute, University of Nevada.

[78] Rigaud, Christophe and Doucet, Antoine and Coustaty, Mickaël and Moreux, Jean-Philippe. ICDAR 2019 competition on post-OCR text correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593. IEEE, 2019.

[79] Rijhwani, Shruti and Anastasopoulos, Antonios and Neubig, Graham. OCR Post Correction for Endangered Language Texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942, Online, November 2020. Association for Computational Linguistics.

[80] Robertson, Stephen and Zaragoza, Hugo. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[81] Rush, Alexander M. and Chopra, Sumit and Weston, Jason. 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.

[82] Schaefer, Robin and Neudecker, Clemens. A Two-Step Approach for Automatic OCR Post-Correction. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 52–57, 2020.

[83] Schnober, Carsten and Eger, Steffen and Do Dinh, Erik-Lân and Gurevych, Iryna. Still not there? Comparing Traditional Sequence-to-Sequence Models to Encoder-Decoder Neural Networks on Monotone String Translation Tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[84] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *COMPUTER NETWORKS AND ISDN SYSTEMS*, pages 107–117, 1998.

[85] Shinoda, Kazutoshi and Aizawa, Akiko. Query-focused Scientific Paper Summarization with Localized Sentence Representation. In *BIRNDL@ SIGIR*, 2018.

[86] Sparck Jones, Karen. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.

[87] Steven Bird and Ewan Klein and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[88] Sutskever, Ilya and Vinyals, Oriol and Le, Quoc V. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[89] Tong, Xiang and Evans, David A. A statistical approach to automatic OCR error correction in context. In *Fourth Workshop on Very Large Corpora*, 1996.

[90] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia. Attention is All you Need. In I. Guyon and U. V. Luxburg and S. Bengio and H. Wallach and R. Fergus and S. Vishwanathan and R. Garnett, editor, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[91] Veličković, Petar and Cucurull, Guillem and Casanova, Arantxa and Romero, Adriana and Liò, Pietro and Bengio, Yoshua. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[92] Wang, Alex and Singh, Amanpreet and Michael, Julian and Hill, Felix and Levy, Omer and Bowman, Samuel. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop Blackbox NLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[93] Wolf, Florian and Gibson, Edward. Paragraph-, Word-, and Coherence-based Approaches to Sentence Ranking: A Comparison of Algorithm and Human Performance. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 383–390, Barcelona, Spain, July 2004.

[94] Yasunaga, Michihiro and Kasai, Jungo and Zhang, Rui and Fabbri, Alexander R. and Li, Irene and Friedman, Dan and Radev, Dragomir R. ScisummNet: A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7386–7393, Jul. 2019.

[95] Yi Tay and Mostafa Dehghani and Samira Abnar and Yikang Shen and Dara Bahri and Philip Pham and Jinfeng Rao and Liu Yang and Sebastian Ruder and Donald Metzler. Long Range Arena: A Benchmark for Efficient Transformers. *arXiv*, cs.LG:2011.04006, 2020.

[96] Yinhan Liu and Myle Ott and Naman Goyal and Jingfei Du and Mandar Joshi and Danqi Chen and Omer Levy and Mike Lewis and Luke Zettlemoyer and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[97] Yonghui Wu and Mike Schuster and Zhifeng Chen and Quoc V. Le and Mohammad Norouzi and Wolfgang Macherey and Maxim Krikun and Yuan Cao and Qin Gao and Klaus Macherey and Jeff Klingner and Apurva Shah and Melvin Johnson and Xiaobing Liu and Łukasz Kaiser and Stephan Gouws and Yoshikiyo Kato and Taku Kudo and Hideto Kazawa and Keith Stevens and George Kurian and Nishant Patil and Wei Wang and Cliff Young and Jason Smith and Jason Riesa and Alex Rudnick and Oriol Vinyals and Greg Corrado and Macduff Hughes and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144, 2016.

[98] Zaheer, Manzil and Guruganesh, Guru and Dubey, Kumar Avinava and Ainslie, Joshua ,and Alberti, Chris and Ontanon, Santiago and Pham, Philip and Ravula, Anirudh and Wang, Qifan and Yang, Li and others. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

[99] Zarinbal, Marzieh and Mohebi, Azadeh and Mosalli, Hesamoddin and Haratinik, Razieh and Jabalameli, Zahra and Bayatmakou, Farnoush. 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.

[100] Zhao, Jinming and Liu, Ming and Gao, Longxiang and Jin, Yuan and Du, Lan and Zhao, He and Zhang, He and Haffari, Gholamreza. SummPip: Unsupervised Multi-Document Summarization with Sentence Graph Compression. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1949–1952, New York, NY, USA, 2020. Association for Computing Machinery.