Leipzig University
Institute of Computer Science
Degree Programme Computer Science, M.Sc.

Citance-Contextualized Summarization of Scientific Papers

Master's Thesis

Ahmad Dawar Hakimi

1. Referee: Jun.-Prof. Dr. Martin Potthast
2. Referee: Asst.-Prof. Dr. Khalid Al-Khatib

Supervisor: M.Sc. Shahbaz Syed

Submission date: August 15, 2023

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, August 15, 2023

.....
Ahmad Dawar Hakimi

Acknowledgments

At this point, I would like to express my sincere gratitude to all the people who have accompanied and supported me throughout my journey to obtain my Master's degree. First and foremost, I would like to express my appreciation to Shahbaz Syed, Khalid Al-Khatib, and Martin Potthast, who supervised me throughout my thesis. Through their guidance, it was possible for me to develop an initial research idea into a well-thought-out thesis. They supported me in crafting the Thesis in all its comprehensive complexity and to create a publication out of it that is currently in the review process. Shahbaz, in particular, has helped me better understand and master the entire process of academic writing and conducting research in general. I appreciate his role not only as a supervisor who always had time for my concerns but also as someone who has become a friend.

The journey also introduced me to the world of academia and would not have been possible if Andreas Niekler and Prof. Gerhard Heyer had not trusted an inexperienced undergraduate student. They led to the fact that I was allowed to work in the NLP group at Leipzig University on various interesting topics and research projects. I met wonderful people in this world who helped me to become a researcher and enriched my life in different facets. Therefore, special thanks go to Christian Kahmann, Christopher Schröder, Kim Bürgl (who all three generously provided me with chocolate during my master's thesis), Janos Borst-Grätz, Soheila Sahami, Erik Körner, Christofer Meinecke, Thomas Efer, and Petra Gamrath. Christian, Erik, Christopher, and Dominik Schwabe were also kind enough to help me with the evaluations and the setup of the LLMs.

Of course, my journey was not only accompanied by my colleagues but also by people I can proudly call my friends. They have been by my side throughout my studies and have made my study experience a unique one. Together we have shared wonderful moments but also persevered through frustrations and challenges. I am deeply grateful for the time I was able to spend with them, and I'm looking forward to all the wonderful moments ahead. They were instrumental in helping me develop both as a researcher and as a human being.

At this point, I would like to express my heartfelt thanks to Akram Sharif, Lena Sophie Voss, Tim Eggert, Kenny Höber, Paul Lukas Naumann, Lukas Gehrke, Lena Rosenbusch, Jakob Kleiber, and Klara Wolf.

Last but not least, I would like to express my deepest gratitude to my family and especially to my father Baryaly Hakimi. Without him, I would not be at the point in my life that I am currently at. He raised me on his own, shared his love with me, planted a strong sense of discipline within me, and nurtured an enduring sense of curiosity. Therefore, I am deeply grateful and would like to say: Taschakor Padar jan!

Abstract

Current approaches for automatically summarizing scientific papers aim to generate informative abstracts that provide a general overview of the paper. However, these abstracts may not fully satisfy the needs of readers who track citations and seek specific connections between the citing and cited papers. In such cases, readers often have to locate the relevant information manually. We propose a novel approach for *contextualized* summarization of scientific papers, which focuses on generating informative summaries that are dependent on the citances (citation texts) in the citing paper. By grounding the summaries in the context of the citances, readers can quickly find the relevant information they seek. Additionally, these contextualized summaries may offer better coverage and focus than generic abstracts. Our approach involves extracting and modeling citances with context, retrieving relevant passages from the cited paper based on citation-context queries, and generating an abstractive summary tailored to the citances. We evaluate our approach on a newly developed high-quality dataset CONTEXT-SCISUMM, comprising 540K papers and 4.6M citances from the computer science domain.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The inception of automatic summarization of scientific works can be traced back to the initial studies in computer science (Baxendale, 1958; Luhn, 1958). Automatically generated abstracts (summaries) were used to create "index volumes" dedicated to specific scientific fields, assisting researchers in managing and navigating the expanding volume of publications. Nowadays, scientific papers typically include abstracts written by the authors themselves. However, such author-generated abstracts may provide incomplete or biased coverage of scientific papers (Elkiss et al., 2008). As a result, the purpose of automatically summarizing papers has evolved to generating more informative summaries, often employing abstractive techniques (Cachola et al., 2020; Cohan et al., 2018; Mao et al., 2022).

A highly practical application of these summaries is to enhance the user's overall *reading* experience. For instance, CiteRead (Rachatasumrit et al., 2022), a part of the Semantic Reader project (Lo et al., 2023), provides an in-situ overview of the cited paper populated by an auto-generated TL;DR summary or its abstract. A scientific paper usually cites several other papers and, in some cases, may even cite a specific paper multiple times, albeit in different *contexts*. In such scenarios, simply using an abstract as the summary may not always be suitable. While abstracts provide a concise and general overview of a paper for potential readers, they may not fulfill the requirements of all readers. Abstracts are helpful for individuals seeking to assess a paper's relevance to their own work (e.g., to find related work). However, they may not sufficiently address the needs of readers who are specifically following a citation. For the latter, an abstract often indicates only the relevance of the cited paper to the citing paper without clarifying how it relates to the specific sentence in which it is cited (hereafter referred to as a *citance*). Since quoting relevant parts of a cited work verbatim is not common practice, manually locating the relevant information is necessary when following a citation.

Hence, we argue that a *contextualized* summary of the cited paper, which is both informative and relevant to the current citance in the paper, would be advantageous.

This thesis investigates the adequacy of abstracts as informative summaries compared to contextualized summaries tailored to each citation. To achieve this objective, we propose a novel approach for generating citance-dependent contextualized summaries of scientific papers (Chapter 3).

Our approach consists of three steps:

1. *extraction and modeling* citances with context from the citing document
2. *retrieval* of relevant content from the cited document using queries based on citance-contexts, and
3. *generation* of abstractive, informative, and citance-contextualized summaries of the cited document.

To address this novel task, we compile CONTEXT-SCISUMM (Chapter 4), a large-scale, high-quality dataset consisting of 540K documents and 4.6M citances from the computer science domain. Through an extensive comparative evaluation using this corpus, we thoroughly examine different variants of our approach compared to the cited paper's abstracts (Section 5.2.2).

1.1 Use Cases

Scientists read scientific publications in various scenarios. A study conducted by Erera et al. (2019) surveyed NLP experts on the frequency and purpose of their reading. The survey included a Ph.D. student, two junior researchers, two senior researchers, and one research strategist. The following are the most common reasons for reading publications among these scientists:

1. keeping track of current work to stay up to date
2. preparing a research project or grant proposal
3. reading relevant papers when writing a scientific paper
4. checking the originality of an idea
5. learning a new field or technology

While scenarios 2 to 5 hold importance, they are not often considered the primary reason for reading publications throughout the year. On the other hand, scenario 1, which involves keeping up with the latest research, is frequently cited as the primary motivation for reading scientific papers daily or weekly.

In numerous situations, having contextualized summaries of scientific papers can be very useful. These summaries aid readers in comprehending research papers more efficiently and accurately and also help them stay up-to-date with current research. This is particularly crucial for individuals new to a field and requiring familiarity with foundational publications.

Contextualized summaries can also be highly beneficial for students, particularly those juggling multiple course projects in a single semester and needing to review literature from different fields. Additionally, people outside of academia who want to read a scientific article on a particular topic without going through foundational publications can also benefit from contextualized summaries, as they offer a more effective way of comprehending the material.

Additionally, contextualized summaries could be beneficial in the review process. This would enable more effective categorization and comprehension of the new research, resulting in higher-quality reviews.

1.2 Thesis Organization

In the following Chapter 2, we will be reviewing the background and related work for scientific summarization. We will first define the terminology for scientific summarization and describe the structure and content of scientific papers. Then, we will introduce the different ways of categorization for summarization approaches (Section 2.3), including generic summarization (Section 2.4) and citation-based summarization (Section 2.5). We will present individual approaches to the summarization of scientific papers and briefly explain their details and the datasets used.

In Chapter 3, we will explain our approach for creating contextualized summaries with the following steps: modeling the citation-context (Section 3.1), formulating the queries (Section 3.2), and citance-guided information retrieval (Section 3.3). The dataset we created will be described in more detail in Chapter 4, including details about the data source, preprocessing, and other statistics. Furthermore, Chapter 5 will present the experiments conducted with the dataset, as well as the automatic and manual evaluations. Finally, Chapter 6 will provide a summary, a critical discussion about the individual parts and contributions of the thesis, and an outlook on further research possibilities for contextualized summarization.

The following Sections: Abstract, 1, 2.5, 3, 3.3, 3.4, 5.2, 5.2.1, Tables: 4.1, 5.1, 5.2, and Figures: 3.1, 5.2 are submitted **verbatim** to an anonymous publication.

The following Sections: 2.4.1, 2.4.3, 2.4.4 are submitted in a **shortened version** to an anonymous publication.

Parts of the following Sections: 2.4.6, 4, 4.4, 3.1.1, 4.1, 4.2, 5.1, 5.2.2, 6, and Table: 5.3 are submitted to an anonymous publication.

Chapter 2

Background & Related Work

In this chapter, we will provide the necessary background information for summarizing scientific papers. We will define domain-specific terms that are sometimes used ambiguously in the literature and describe the most common structure of a scientific paper, along with the information expected in each section.

We then explain the notion of summarization and provide an overview of the different ways to categorize summarization algorithms. For this thesis, we will focus on the most relevant categorization based on context, which distinguishes between generic, citation-based, and update summaries. However, since update summaries are not within the scope of our type of summaries, we will only present the current approaches and state of research for generic and citation-based summarization. Additionally, we will briefly discuss the datasets used and their details. Finally, we will highlight the differences between our approach to creating contextualized summaries and our dataset CONTEXT-SCISUMM compared to previous approaches and datasets.

2.1 Terminology

Within the literature, certain subject-specific terms are used ambiguously. To ensure accuracy and clarity within this thesis, we have provided explicit definitions for each of these terms. Some of our definitions were sourced from Altmami and Menai (2022)'s survey, while others were created specifically for this paper to define other domain-specific terms.

Scientific Paper: A document presenting an original scientific study's methodologies, findings, and conclusions. It adheres to a specific structure that includes standardized sections, such as the abstract, introduction, methodology, results, and discussion.

Citing Author: The author who cites, discusses, or mentions the results or methods of another research study in his or her work.

Citing Paper: A scientific paper that contains a direct citation to another publication.

Cited Paper: A scientific paper referenced by a citing paper.

In the literature, the following terms are used interchangeably. In this thesis, we will use the term *citance*:
Citance / Citation Sentence / Explicit Citation-Context (ECC):
A sentence that contains a citation marker or reference to another publication. It acknowledges the source of the target information, supports the author's argument, and enables readers to locate the cited information in the cited paper.

Implicit Citation-Context (ICC) / Citation-Context: Sentences in the citing document that do not contain the exact citation marker as the citance but share semantic similarities with the citance or provide supplementary information.

Cited Text Span / Relevant Content: The specific portion of text in a cited paper that is referred to by a citance.

2.2 Structure of a Scientific Paper

Scientific papers do not have a strictly fixed structure but generally follow a specific scheme. This scheme includes an abstract, introduction, related work, methodology, experimental section, discussion, conclusion, and references (Altmami and Menai, 2022).

Abstract: is a brief, informative summary of the main aspects of the publication, including the objectives, materials, methods used, and results and conclusions, typically consisting of 150-250 words. It should answer the following questions:

- Why was the study conducted?
- How was it conducted?
- What conclusion was drawn?

Introduction: provides necessary background information about the problem being addressed, its significance, and the objectives of the work. It should also include an operational definition of the terms used and is typically 300-500 words in length.

Related Work: summarizes the literature on the problem being addressed to bring the reader up to date with current research and present one's research in the context of existing literature.

Methodology: this section describes the details of the procedure or algorithm used or developed, which are needed to reproduce the author's work and results.

Experimental Section: this section describes the details of the experiments, including outcomes, results, and various statistics, among other things.

Discussion: in the discussion, the results of the experiments are evaluated and explained, and their implications for future research on this problem are described.

Conclusion: provides a general summary of the document, including the implications and findings, and usually includes suggestions for future research directions.

References: is the last section and consists of listing all referenced scientific papers within the document.

2.3 Types of Summarization

The goal of summarization algorithms is to condense a text's important information to make it quicker and easier to understand. There are different ways to categorize these algorithms, including by their *function* (indicative or informative), *type* (extractive or abstractive), *source* (news, scientific, literary, social media), *number of documents*, and *context* (generic, citation-based, or update (Torres-Moreno, 2014).

Indicative summaries provide information about the citing document's topics, while informative summaries reflect the text's content and main arguments. Extractive approaches compile fragments from the citing document, while abstractive algorithms summarize the text by rephrasing and rearranging sentences for better comprehension.

Different types of texts require different summarization approaches, including consideration of their length and characteristics. Depending on the context, algorithms can provide a generic overview, consider the user's specific information needs, or show new information without repeating old information.

We create informative, abstractive, and context-based summaries for our contextualized summaries. Our focus is, therefore, on generic and citation-based summaries of scientific papers. Firstly, we review the literature and then discuss the different types of generic summarization approaches. Next, we present approaches for citation-based summaries that use the citation-context or citance from the citing paper to summarize the cited paper.

2.4 Generic Summarization

In the following sections, we will discuss various methods that belong to the category of generic summarization. These summaries aim to provide a neutral summary of a scientific paper without catering to the specific information requirements of the user. The summary gives an overall idea of the publication and is a concise version of the original document.

2.4.1 Abstract-based Summarization

For learning-based approaches, a supervised model for extractive summarization by Collins et al. (2017) was trained on a corpus of 10,148 computer science papers, where each paper included both author-provided highlights and an abstract that served as the reference summary. Additional training data was generated using ROUGE (Lin, 2004) by extracting sentences from the document that had significant overlap with the highlights. The approach involved training an LSTM-based neural encoder along with several lexical features to classify sentences as summary-worthy. Another approach by Cohan et al. (2018) focused on discourse-aware attention modeling for abstractive summarization of scientific papers from the arXiv (215K) and PubMed (133K) collections. This model used a hierarchical encoder to capture the discourse structure of the document, while the decoder considered both the section information and the previously generated tokens to ensure a coherent abstractive summary. Furthermore, Gupta et al. (2021) conducted a study investigating the advantages of pre-training and fine-tuning BERT-based models for extractive summarization.

2.4.2 Table-of-content and Lay Summarization

In contrast to approaches that generate summaries resembling abstracts, Chen et al. (2020) produced table-of-contents summaries for journal articles, where short author-written advertising blurbs were considered as the ground truth summary. Besides, Chandrasekaran et al. (2020) introduced the task of lay summarization, which aims to generate simple and accessible summaries of scientific papers for non-experts. They created a corpus of 572 documents along with author-generated lay summaries to facilitate research in this area. Also, Zaman et al. (2020) combined text simplification and summarization techniques to generate layman summaries for 5204 scientific papers. They utilize news articles that describe a scientific paper as a reference for creating layman summaries. Guo et al. (2021) delivered a corpus of 7805 abstracts of systematic reviews paired with their plain language versions written by domain experts. Goldsack et al. (2022) introduced two larger datasets for lay summarization, including summaries with varying degrees of readability to serve a diverse audience.

2.4.3 TL;DR Summarization

Unlike the previously discussed summaries that output informative and paragraph-sized texts, ultra-short TL;DR (too long; didn't read) style summaries are concise (e.g., one or two sentences). These summaries serve as indicative summaries, aiming to highlight the most important finding from the document. In this regard, Cachola et al. (2020) generated TL;DR summaries via control codes and multi-task learning. They developed the SCITLDR corpus that comprises 3.2K documents, each accompanied by a manually written TL;DR formulation (15-25 words long) of the summary. These TL;DR summaries were derived from the summary provided by peer reviews, usually found in the first paragraph, as well as from the author of the document. As an auxiliary training signal, their model also generates the title given an abstract.

2.4.4 Long Informative Summarization

The LongSumm task (Chandrasekaran et al., 2020) focuses on generating longer summaries of approximately 600 words. This task is motivated by the understanding that neither abstracts nor TL;DR summaries provide sufficient information to serve as a substitute for reading the cited paper. As a result, various datasets and models have been proposed to address the challenge of generating comprehensive summaries for scientific papers. For example, Chandrasekaran et al. (2020) created the LongSumm corpus, which consists of 2258 documents with abstractive and extractive summaries. Sotudeh et al. (2021)

created two corpora sourced from arXiv and PubMed containing 11,149 and 88,701 document-summary pairs, respectively. Sotudeh and Goharian (2022) extended the abstracts by using *introductory* sentences (those from the *introduction*, *overview*, and *motivation* sections) to guide the long summary generation.

2.4.5 Multi-faceted and Argument-based Summarization

Meng et al. (2021) proposed *faceted* summarization of scientific papers where they generated multiple summaries for some facets of the document such as *purpose*, *method*, *findings*, and *value*. Soleimani et al. (2022) also used section titles as aspects in a zero-shot summarization setting. Teufel and Moens (2002) introduced *argumentative zones*, which classify sentences on the basis of their rhetorical status in the discourse, such as *aim*, *background*, *basis*, *contrast*, and *own* (contributions). The sentences belonging to the *aim* and *own* zones were leveraged to create a relevant paper summary highlighting its novel contributions. Argumentative zoning was also employed by Contractor et al. (2012) for structured summarization of biomedical articles. Several zones such as *background*, *hypothesis*, *goal*, *method* were annotated by Liakata et al. (2013) and used to identify key sentences for creating an extractive summary.

2.4.6 Our Summaries

go beyond generic summarization by delivering multiple summaries that are relevant to specific citance contexts within a single scientific paper. Regarding the length of the summaries, they are shorter compared to the ones proposed by LongSumm, but longer than TL;DR summaries. Also, our approach incorporates retrieval models to extract pertinent information from the cited paper, allowing us to provide inputs of varying lengths to the summarization model. This helps control the desired level of detail in the output summaries, effectively adjusting the granularity according to certain requirements. The entire approach to generating contextualized summaries is described in detail in Chapter 3.

2.5 Citation-based Summarization

Citation-based summarization utilizes citances from the citing paper to extract key information from the cited paper and generate a summary. In their work, Qazvinian and Radev (2008) analyzed the citation network of the cited paper, collecting citances from different citing papers. These citances were clustered, and central sentences were identified as the extractive summary. On

the other hand, Mei and Zhai (2008) focused on impact-based summaries, retrieving impact sentences that reflect the citation's authority and proximity in a collection. The impact of the cited paper on related work was determined through citations from citing papers. To enhance readability and coherence, Abu-Jbara and Radev (2011) introduced a preprocessing stage to filter out irrelevant fragments by tagging the scope of the target reference. They then performed an extraction stage to select important sentences from sections such as background, problem statement, method, results, and limitations. Finally, a post-processing stage improved overall readability by replacing pronouns and resolving co-references.

Closely related to our work, Cohan and Goharian (2015) employed citation-contexts, defined as the textual spans from the cited paper that reflect the citation from the citing paper. To summarize the cited paper, they first collected multiple citation-contexts and constructed a graph based on their intra-connectivity (cosine similarity of the `tf-idf` vectors). Sentences within this graph were ranked based on their importance (number of connections). These important sentences were combined with the discourse information of the cited paper to create an informative summary. This model was further improved by leveraging word embeddings and domain knowledge to enhance the citation-contexts (Cohan and Goharian, 2017).

While our work shares a focus on contextualizing citations, there is a key difference in our approach. In particular, we formulate different types of citation-contexts extracted from the citing document and use them to generate multiple context-relevant summaries for a given cited paper. Notably, instead of relying solely on using the citance verbatim as the query, representing only one type of citation-context, we examine the utilization of various citation-contexts as queries. Besides, we employ multiple retrieval models to extract pertinent information from the cited paper, enhancing the relevance of the generated summaries (Chapter 3).

Regarding the data, our corpus ContextSciSumm (Chapter 4) is the largest for citation-context specific summarization of scientific papers, containing around 540,000 documents and 4.6 million citances. In comparison, the CITESUM corpus by Mao et al. (2022) is smaller with 93,000 documents, where the citance text from the related work section of the citing paper serves as an ultra-short summary of the cited paper. Our corpus encompasses citances from all sections of the citing paper and includes multiple types of citation-contexts and corresponding summaries for each context. As a result, our dataset offers a more comprehensive and diverse resource for studying scientific paper summarization.

Chapter 3

Citance-contextualized Summarization

Our approach to contextualized summarization involves leveraging multiple contexts of the citances in a citing paper. In addition to the citance itself (a single sentence containing the citation marker), we consider various types of surrounding contexts. As illustrated in Figure 3.1, our approach comprises three main steps: (1) extraction and modeling of citances, (2) retrieval of relevant information from the cited paper, and (3) generation of abstractive summaries that are contextualized based on the citance. We provide a detailed description of each step below.

3.1 Modeling Citation-Contexts

A crucial step is to link a citance to the relevant section of the cited paper, which is known as *Citation Contextualization*. This process involves identifying the appropriate context within a cited paper for each citance to be extracted. Citances are usually inadequate because they only provide a brief summary of the referenced text, leaving out many details. References to a paper's contributions are often made without providing necessary background information on the data used and how they obtained the results (Altmami and Menai, 2022). Additionally, there may be inconsistencies between the terminology used by the citing authors and the cited authors (Cohan and Goharian, 2018), making it difficult to contextualize citations properly. To address these issues, we use different modeling approaches to the citation-context. According to the survey papers by Rotondi et al. (2018) and Álvarez and Gómez (2016), there are three methods for modeling the citation-context:

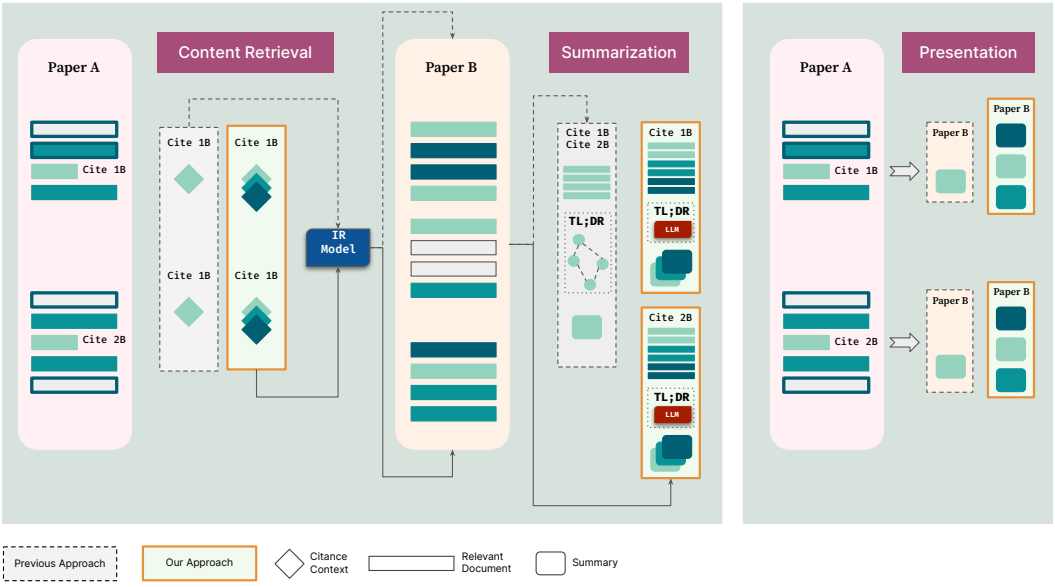

Figure 3.1: Comparison of previous approach Cohan and Goharian (2015) with our approach to contextualized summarization of scientific papers. Our approach considers multiple citation-contexts from the citing paper for each citance (Cite 1B/2B in Paper A) to retrieve relevant content from the cited paper (Paper B). Multiple summaries are generated from the retrieved content, each tailored to a specific citation-context, as shown in the presentation. In contrast, the previous approach generates a single summary for all the citances to Paper B by aggregating relevant content for all citances.

1. *citance*
2. *fixed number of characters* preceding and succeeding the citation marker
3. *Extended Context/ Implicit Citation-Context* tries to find or classify sentences that can provide additional information to the citance within the citing paper.

However, straightforward approaches to (2), which extract a fixed number of characters before and after the citation marker, can result in truncated sentences. On the other hand, more complex approaches are limited to supervised methods (Álvarez and Gómez, 2016). For this reason, we used (1) the citance verbatim as a query and (3) two different types of implicit citation-context, which will be described below.

3.1.1 Implicit Citation-Context

We begin by extracting all citances verbatim from a given paper, which refers to other papers. Then, we examine two additional contexts for each citation. The first context encompasses the sentences immediately preceding and following the citance. The second context involves the two most similar sentences in meaning to the citance within the same paragraph. In total, we have three citation-contexts: ***citance*** (citance itself), ***neighbors*** (citance and neighboring sentences), and ***similar*** (citance and semantically similar sentences). By incorporating these contexts, we aim to enhance the exploration of relevant information from the cited paper, thereby facilitating a deeper comprehension of the underlying rationale behind the citance.

Neighbors Approach

To apply the neighbors approach, we extract the preceding and following sentences as implicit citation-context for a citance. This approach is based on the analysis of Qazvinian and Radev (2010). The authors manually labeled sentences from 10 publications of the ACL Anthology Network (Radev et al., 2013) as *explicit citation*, *context sentence*, or *none*. They then applied a Markov random field model to determine the context sentences. Their findings revealed that most implicit citation-context sentences have a gap size of 0 to the citance and therefore are located around the citance. However, since the corpus they used was relatively small and the analysis showed that some implicit citation-context sentences had a gap size greater than 0, we adopted a second approach to model citation-context.

Similar Approach

The similar approach involves identifying the most semantically similar sentences to a citance within the same paragraph. To achieve this, we utilized SciBERT (Beltagy et al., 2019) to embed both the citance and the sentences in the surrounding paragraph. SciBERT is a pre-trained Bert model designed for scientific texts, and it can generate meaningful embeddings in our case. We applied cosine similarity to determine the two most similar sentences.

3.2 Query Formulation

In the second step of our approach to generating contextualized summaries, we utilize the modeled citation-contexts as queries to extract the relevant content from the cited paper. We also utilize the extracted keywords of the three

queries - *citance*, *neighbors*, and *similar*, separately. Below, we provide a description of the keyword extraction process.

3.2.1 Keyword Extraction

We use KeyBERT (Grootendorst, 2020) combined with SentenceTransformers *all-mpnet-base-v2*[1] and *SciBERT* (Beltagy et al., 2019) to extract keywords from queries. Firstly, the Transformer Models extract document embeddings to obtain a document-level representation. After that, word embeddings are extracted for n-gram words/phrases. Finally, it uses cosine similarity to identify the words/phrases that are most similar to the document. The words that are most similar can then be identified as the words that best describe the entire document. The only problem with this is that KeyBERT requires the user to specify an n-gram range, but the optimal length is unknown, resulting in truncated keyphrases. To overcome this problem, we use KeyphraseVectorizers (Schopf et al., 2022). It uses the part-of-speech tags of a document, determines their frequency, and extracts extended noun phrase keyphrases. These keyphrases have a structure that includes an article, one or more adjectives, and a noun.

An example of the extraction process for the citance is provided below:

Citance:
For the question representation, since a well-formed question might sensitive to the word order, we make use of the recent proposed contextual word embeddings such as BERT [7] to capture the contextual word information.

Neighbors:
Previous Sentence:
To overcome the aforementioned problems, we propose a Seq2Seq model for question refinement, referred to as QREFINE.

Next Sentence:
The utilization of BERT, which is trained on a large-scale unlabeled corpus, also helps alleviate the issue of data sparsity, particularly when there is insufficient training data available.

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Similar:
Previous Sentence: ""

Next Sentence:
Furthermore, since ill-formed questions might contain typographical errors, we augment the question representation with fine-grained character embeddings [24]. To enhance the generation of desired questions, we introduce a training algorithm based on reinforcement learning for the Seq2Seq model.

Step 1: Extract the keyphrases with *all-mpnet-base-v2* and *SciBERT*.

- *keywords-SciBERT*:

```
[
    ('contextual word information', 0.6971),
    ('contextual word embeddings', 0.6706),
    ('question representation', 0.6557),
    ('word order', 0.5261),
    ('use', 0.2681)
]
```

- *keywords-all-mpnet-base-v2*:

```
[
    ('question representation', 0.6747),
    ('contextual word embeddings', 0.5889),
    ('word order', 0.3886),
    ('bert', 0.3142),
    ('use', 0.081)
]
```

Step 2: Combine the lists of keyphrases and only keep one occurrence of duplicates while retaining the highest score from each list.

- *keywords-merged*:

```
[
    ('contextual word information', 0.6971),
    ('contextual word embeddings', 0.6706),
    ('question representation', 0.6747),
    ('word order', 0.5261),
    ('use', 0.2681),
    ('bert', 0.3142)
]
```

Step 3: Calculate the mean score.

- mean score: 0.525

Step 4: Filter out any keyphrases that have a score lower than the mean score.

– *keywords-final*:

```
[
    ('contextual word information', 0.6971),
    ('contextual word embeddings', 0.6706),
    ('question representation', 0.6747),
    ('word order', 0.5261)
]
```

These are the final keyphrases for the citance in the three different variants:

citance	**neighbors**	**similar**
contextual word information	contextual word information	contextual word information
contextual word embeddings	contextual word embeddings	contextual word embeddings
question representation	question representation	question representation
word order	word order	word order
	seq2seq model	seq2seq model
	qrefine	baselines
	enough training data	
	data sparsity problem	
	bert	

Table 3.1: Overview of the keyphrases extracted for the three citation-contexts, citance, neighbors, and similar. The keyphrases were extracted using the four steps outlined in Section 3.2.1.

3.2.2 Overview Queries

In total, we used 12 different retrieval scenarios to extract the relevant content from the cited paper. To do this, we used three different citation-contexts: *citance*, *neighbors*, and *similar*. Additionally, we considered the keywords separately. Then, we used both a shallow retrieval model, specifically BM25, and a dense retrieval model, utilizing SciBERT. An overview and a detailed description of all retrieval scenarios can be found in Table 3.2.

Retrieval Scenarios	Description
`citance-BM25`	Citance used as the query, documents ranked with BM25.
`citance-SciBERT`	Citance used as the query, documents ranked based on cosine similarity of SciBERT embeddings.
`citance-keywords-BM25`	Keywords extracted from citance used individually as queries, documents ranked with BM25. Ranked lists are then fused into a final ranking.
`citance-keywords-SciBERT`	Keywords extracted from citance used individually as queries, documents ranked based on cosine similarity of SciBERT embeddings.
`neighbors-BM25`	Citance and surrounding sentences (i.e., the *neighbors* context) used as the query, documents ranked with BM25.
`neighbors-SciBERT`	The *neighbors* context used as the query, documents ranked based on cosine similarity of SciBERT embeddings.
`neighbors-keywords-BM25`	Keywords extracted from the *neighbors* context used individually as queries, documents ranked with BM25. Ranked lists are then fused into a final ranking.
`neighbors-keywords-SciBERT`	Keywords extracted from the *neighbors* context used individually as queries, documents ranked based on cosine similarity of SciBERT embeddings. Ranked lists are then fused into a final ranking.
`similar-BM25`	Citance and two semantically similar sentences in the same paragraph (i.e., the *similar* context) used as the query, documents ranked with BM25.
`similar-SciBERT`	The *similar* context used as the query, documents ranked based on cosine similarity of SciBERT embeddings.
`similar-keywords-BM25`	Keywords extracted from the *similar* context used individually as queries, documents ranked with BM25. Ranked lists are then fused into a final ranking.
`similar-keywords-SciBERT`	Keywords extracted from the *similar* context used individually as queries, documents ranked based on cosine similarity of SciBERT embeddings. Ranked lists are then fused into a final ranking.

Table 3.2: Overview of the various retrieval scenarios to find relevant content from the cited paper. Documents may be sentences or paragraphs. For the keyword-based scenarios, the final ranking is obtained by a weighted aggregation of the rankings based on the cosine similarity of a keyword and the citance.

3.3 Citance-guided Information Retrieval

We utilize the three citation-contexts mentioned earlier as queries to retrieve pertinent information from the cited paper. Additionally, we explore the use of extracted keywords from each citation-context to enhance the queries (Carpineto and Romano, 2012). For retrieval, we employ both shallow and dense retrieval models (Section 4.2). To extract the relevant content, we first remove the citation marker from the query. Additionally, we do not retrieve sentences from the abstract and conclusion as they contain summary sentences. We retrieve relevant content at two levels of granularity: sentences and paragraphs. Specifically, we extract the top-5 relevant sentences and the top-2 relevant paragraphs from the cited paper. This enables us to assess which granularity is more suitable for the contextualized summarization task.

The top-5 relevant sentences provide a broader *coverage* of the cited paper, encompassing diverse information that is relevant to the citance. Conversely, the top-2 relevant paragraphs offer a higher degree of *focus*, where the summary sentences are interconnected. Hence, we experiment with both granularities to explore their efficacy in our approach. Following the retrieval process, we qualitatively evaluate the retrieved content. This evaluation (Section 5.1.1) aids us in selecting the optimal combination of query and retrieval models for the subsequent summarization step (Section 5.2).

3.4 Citance-contextualized Summarization

After retrieving the relevant content from the cited paper, we utilize it as input for the summarization model. This ensures that the generated summaries are grounded in the context of the citance, focusing solely on the parts of the cited paper that are relevant to the citance. Our approach explores the effectiveness of large language models (LLMs) due to their strong multi-task capabilities (Bommasani et al., 2021). Specifically, we employ prompt-based, instruction-following models that can understand and execute natural language instructions provided by the user to accomplish a given task. This flexibility and adaptability to various domains distinguish our approach from domain-specific supervised methods.

Since we have two granularities of input for the summarization model (top-5 sentences and top-2 paragraphs), we designed two prompts tailored to each. For the top-5 sentences, we employ a paraphrasing prompt that aims to transform the set of sentences into a coherent summary. On the other hand, for the top-2 paragraphs, we use an abstractive summarization prompt to generate a cohesive summary from the set of paragraphs. Further details regarding the prompts can be found in Section 5.2.1.

Chapter 4

CONTEXT-SCISUMM: A Large-scale Corpus for Contextualized Summarization of Scientific Papers

Previous datasets for scientific paper summarization do not account for different types of citation-contexts, evaluate multiple retrieval models for extracting relevant content, or use citances from every section of the paper (Chapter 2). Therefore, these datasets are unsuitable for studying citance-contextualized summarization in their original form. To address this gap, we introduce CONTEXT-SCISUMM, a new and extensive dataset created using our approach described in Chapter 3. It comprises 540,000 documents and 4.6 million instances, making it the most extensive dataset for scientific summarization.

Furthermore, this chapter provides information on the underlying data resource and individual preprocessing steps in Section 4.1, the creation of indexes for retrieving relevant content in Section 4.2, the description of an instance in Section 4.3, as well as corpus statistics in Section 4.4.

4.1 Data Source and Preprocessing

We utilized the publicly available Semantic Scholar Open Research Corpus (S2ORC) (Lo et al., 2020) for our study.[1] The corpus comprises 136M scientific papers, of which 12 million have full-text available. This structured dataset is automatically annotated with citations, figures, tables, and links to corresponding publications, totaling 380.5 million resolved citation links.
To create CONTEXT-SCISUMM, we filtered the S2ORC dataset by using only the 12 million publications with full text available and excluding all publi-

[1]We used the S2ORC dataset released on 2020-07-05.

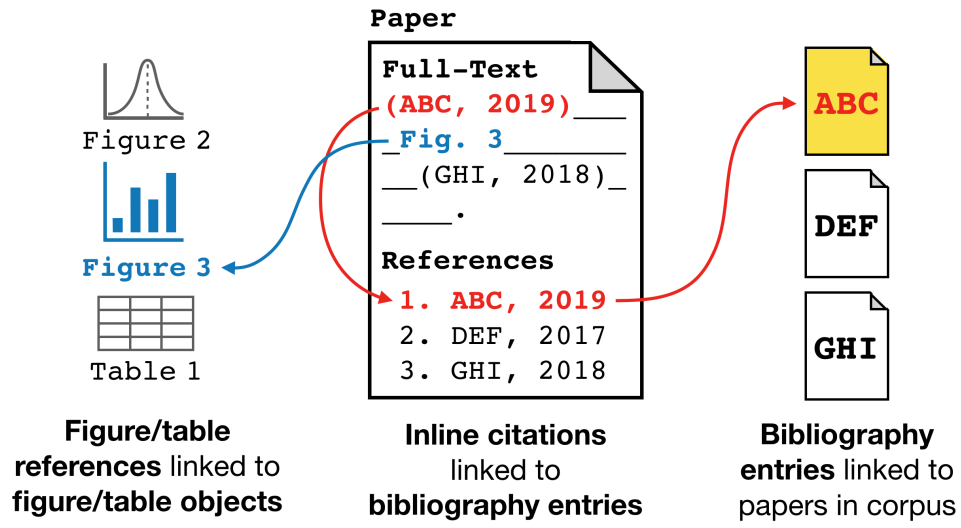

Figure 4.1: Inline citations and references to figures and tables are annotated in S2ORC's structured full text. Citations are linked to bibliography entries linked to other papers in S2ORC. Figure and table references are linked to their captions (Lo et al., 2020).

cations that were not solely computer science papers. We removed all documents where the attribute *mag_field_of_study* did not have exclusively "*CS*" (computer science) as a value. Further, we also filtered out interdisciplinary publications such as those from computational psychology, computational environmental science, and computational art. Our filtering steps resulted in 870,810 documents, which we indexed by sentence and paragraph. We focused on a subset of 870,810 documents from the computer science discipline to ensure the relevance of our evaluation (Section 5.2.2). After removing documents without citations within the corpus, we were left with approximately 540,000 documents. We then extracted citances by identifying the exact sentence containing the citation marker in each document. This resulted in a total of 4.6M citances in our corpus. Unlike Mao et al. (2022), who considered only citances from the *Related Work* section, we retained citances from all sections of a document, resulting in a more diverse set of citances.

4.2 Citation-contexts and Retrieval Models

We devised three citation-context types as queries to retrieve relevant content from the cited paper, as described in Section 3.1. The *citance* and *neighbors* contexts were derived straightforwardly. For the *similar* context, we employed cosine similarity of the contextual embeddings from SciBERT (Beltagy et al.,

2019)[2] to identify the two most semantically similar sentences to the citance. Additionally, we experimented with the keywords extracted from the citance and the citation-context using KeyBERT (Grootendorst, 2020) as individual queries.

For retrieval, we explored BM25 (Robertson et al., 1994)[3] and cosine similarity of the SciBERT embeddings between the query (citation-context) and the document (sentences or paragraphs of the cited paper). This allowed us to examine both shallow and dense retrieval paradigms. The combination of the three query types (including keyword variants) and the two retrieval models resulted in a total of 12 retrieval scenarios, as outlined in Table 5.1, accompanied by their mean nDCG@5 scores from our internal evaluation (Section 5.1).

The 870,810 filtered documents from the S2ORC dataset were already in paragraph form. We split the sentences using the SciSpacy model *en_core_sci_lg* to index the documents at the sentence level. This model was trained on biomedical data and recognized the structure of scientific papers and different citation styles better than the standard Spacy model *en_core_web_lg*. As a result, we obtained better sentence-splitting. We indexed 151M sentences and 40M paragraphs to retrieve the top-5 sentences and top-2 paragraphs, respectively, for each query. Regarding the keyword queries, we merged their individual rankings using weighted aggregation to obtain a final ranking for a single citance. Specifically, each ranking from a keyword query was scaled by its cosine similarity with the citance, and the resulting rankings were aggregated through a weighted sum.

[2] `scibert-scivocab-uncased` from `https://huggingface.co/allenai/scibert_scivocab_uncased`

[3] We used the *Rank-BM25* toolkit (Brown, 2020) with default parameters for BM25 (k=1, b=0.75).

The weighted rank aggregation equation is given by:

$$\text{FinalRanking} = \sum_{k \in K} (s_k * \mathbf{R}_k) \tag{4.1}$$

Where:

- $*$ represents the scaling of the ranking vector $\mathbf{R}_k$ by the cosine similarity s_k
- K is the set of keyword queries
- $\mathbf{R}_k$ is the ranking vector obtained from the k-th keyword query in K
- s_k is the cosine similarity between the citance and the keyword query k

4.3 Data Entry Example

Our corpus consists of approximately 540,000 documents and 4.6 million citances. For each document, we generated multiple files. The citance file (Fig. 4.2) provides information on all the citances included in a citing paper, while the content file contains the extracted contents for the top-performing retrieval scenarios for each document. At present, we have only produced summary files for the papers from the evaluation dataset (Section 5.2.2), as time and cost constraints prevented us from generating multiple contextualized summaries for all documents.

The following information pertains to a single citance extracted from a file that contains multiple such entries. Each citance is assigned a unique identifier referred to as *citance_no*. Additionally, it includes metadata about the citing paper, including *citing_paper_id*, *title*, and *citing_paper_authors*. This is followed by details about the citance and the corresponding cited paper. The *reference* field indicates the citation marker used, while *citance_section* specifies the name of the section where the citance originated. The following two fields, *prev_sentence* and *next_sentence*, contain lists of sentences that provide the implicit citation-context for the citance. After this citation-context is determined, the position in the text is used to determine whether the sentences precede or follow the citance, which is important for our similar approach (Section 3.1.1). Lastly, the file contains metadata for the cited paper, namely *reference_paper_title* and *reference_paper_link*, which includes the linked paper ID. These two pieces of information enable us to retrieve the cited paper from our index.

```
[
  {
    "citance_No": 5,
    "citing_paper_id": 51871042,
    "title": "DeepPavlov: Open-Source Library for Dialogue
        Systems",
    "citing_paper_authors": "D Jason, Kavosh Williams,
       Geoffrey Asadi, Zweig",
    "reference": "(Williams et al., 2017)",
    "citance_section": "Implemented Models and Skills",
    "citance": "The skill implements Hybrid Code Networks
       (HCNs) described in (Williams et al., 2017).",
    "prev_sentence": [
      "Some of them are available for interactive online
         testing."
    ],
    "next_sentence": [
      "The model is configurable: embeddings, slot filling
          component and intent classifier can be switched
         on and off on demand."
    ],
    "reference_paper_title": "Hybrid code networks:
       practical and efficient end-to-end dialog control
       with supervised and reinforcement learning",
    "reference_paper_link": "13214003"
  }
]
```


Figure 4.2: Illustration of a single citance example extracted from a JSONlines file containing multiple examples, showcasing a citance, meta-information about the citing and the cited paper, and the citation-context.

We compiled two different content files for the citances (Appendix A.1, A.2), one for the top-2 and one for the top-5 scenario. These content files consist of several fields. Firstly, the field labeled *citance_no* indicates the corresponding citation number. Next, we have the field *citing_paper_id*, which specifies the paper ID of the citing paper. Following these fields are *similar-BM25*, *similar-keywords-BM25*, *citance-SciBERT*, and *citance-BM25*, which hold the extracted context based on the best-performing retrieval scenarios. For more detailed information about each retrieval scenario, please refer to Table 3.2.

The summary files are in CSV format, but to avoid redundancy, we are referring to them here in their representation as a Table: 5.4, 5.5, and Appendix

B.1. The file contains the following entries: *citing_paper_title* and *reference_paper_title*, which reflect the titles of the citing and cited papers. Then, it contains information about the *citance*, *next_sentence* and *prev_sentence*, where the sentences of the similar approach are stored as implicit citation-context based on the evaluation results of Section 5.1 for the citance. For each document, there is also the *abstract* of the cited paper and the four contextualized summaries: *GPT-citance-SciBert-top5*, *GPT-similar-BM25-top2*, *LLaMA-CoT-citance-SciBert-top5*, and *Vicuna-similar-BM25-top2*.

4.4 Corpus Statistics

The compiled corpus, consisting of 537,155 scientific papers in the computer science domain, encompasses a total of 4,619,552 citances. On average, each paper contains 8.6 citances. The mean length of a citance is 31 tokens, with a median length of 27 tokens (see Fig. 4.3). The citances show a normal distribution around this mean value. There are 4767 outlier values with a citance text length of over 200 tokens, with a maximum length of 5388 tokens. These outliers mainly comprise reviews of papers, and an example can be found in Appendix D.

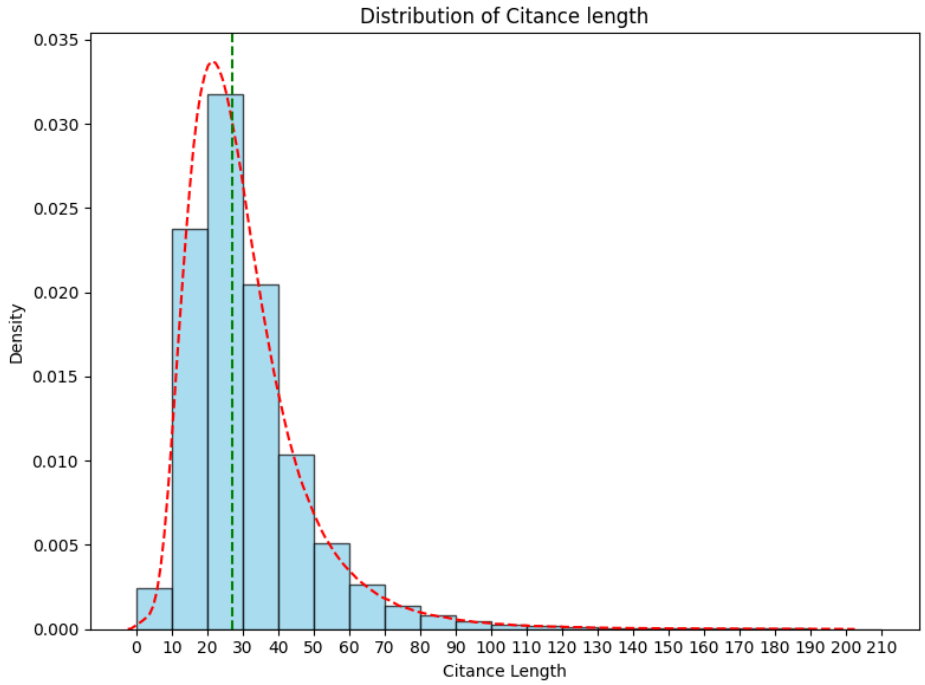

Figure 4.3: Histogram density plot showing the distribution of citance lengths up to 200 tokens. The median length (marked in green) is 27 tokens.

Furthermore, the corpus includes 346,450 papers that feature multiple citances to the same cited paper, making them an optimal subset for studying contextualized summarization approaches. In Table 4.1, we comprehensively compare our corpus with other datasets. We have crafted a unique dataset with multiple, abstractive, citance-contextualized, informative summaries of a cited paper. The average summary length is 117 tokens, calculated on the 100 papers from the evaluation dataset (Section 5.2.2).

Corpus	Size	Length (avg.)	Style	Type	Multiple
LaySumm (Chandrasekaran et al., 2020)	572	84 tokens	Layman-Summary	Abs.	✗
SciSummNet (Yasunaga et al., 2019)	1K	151 words	Abstract	Ext.	✗
TalkSumm (Lev et al., 2019)	1.7K	965 words	Informative	Ext.	✗
LongSumm (Chandrasekaran et al., 2020)	2.2K	779 words	Blog Post	Ext./Abs.	✗
SciTLDR (Cachola et al., 2020)	3.2K	21 words	TL;DR	Abs.	✓
FacetSum (Meng et al., 2021)	60K	290 words	Facet-oriented, Informative	Ext./Abs.	✓
CiteSum (Mao et al., 2022)	93K	23 words	TL;DR	Abs.	✗
CORD-SUM (Qi et al., 2022)	123K	223 words	Abstract	Ext.	✗
PubMed (Cohan et al., 2018)	133K	203 words	Abstract	Abs.	✗
ArXiv (Cohan et al., 2018)	215K	220 words	Abstract	Abs.	✗
RSCSum (Chen et al., 2020)	308K	29 words	Table of Contents	Abs.	✗
ArXiv-Long (Sotudeh and Goharian, 2022)	11.1K	574 tokens	Extended Abstract	Ext.	✗
PubMed-Long (Sotudeh and Goharian, 2022)	88K	403 tokens	Extended Abstract	Ext.	✗
CONTEXT-SCISUMM	540K	117 tokens	Citance-oriented, Informative	Abs.	✓

Table 4.1: Comparison of our CONTEXT-SCISUMM corpus with existing corpora for scientific paper summarization. Shown are the size of each corpus, average summary length, the target style of the summary, summary type, and if multiple summaries are provided per document. Our corpus provides a unique combination of multiple, abstractive, citance-contextualized, informative summaries of a cited paper. Average summary length is over the 100 references created for qualitative evaluation (Section 5.2.2).

Chapter 5

Experiments & Evaluation

This chapter will discuss the experiments and evaluations conducted to develop CONTEXT-SCISUMM and assess its quality and usefulness. Initially, a pilot study was performed to identify the optimal retrieval scenario for extracting relevant contextual information from a citance. Subsequently, we employed a shallow and a dense retrieval model to extract relevant content from the best-performing and paradigmatically different models. To summarize the extracted content, we utilize large language models (LLMs) with different instructions and templates for prompts. A detailed breakdown of the LLMs and prompts can be found in the corresponding Sections 5.2, 5.2.1. Finally, human and automatic evaluations were conducted to determine the usefulness and quality of the generated contextualized summaries (Section 5.2.2).

We needed to conduct two separate evaluation steps to evaluate the generation of contextualized summaries effectively. It was not sufficient to evaluate only the summaries, as this would not allow us to determine whether any inadequacies were due to the summarization model or the retrieving of the relevant content.

5.1 Contextualized Retrieval

Experimental Details: We conducted an internal evaluation via manual relevance judgments of the 12 retrieval scenarios presented in Table 5.1. Specifically, we determined the relevance of the retrieved content from the cited paper to the corresponding citation-context (query) from the citing paper. We employed ten queries to retrieve the top-5 sentences from the cited papers for each of the 12 scenarios, resulting in a total of 600 sentences. These sentences were evaluated for relevance using a three-point scale: *relevant*, *somewhat relevant*, and *non-relevant*. The evaluation metric used was NDCG@5 (Järvelin and Kekäläinen, 2002), as displayed in Table 5.1. This metric measures the

quality of ranked search results. It evaluates the ranking accuracy and relevance of the top 5 items in a list using user feedback. A higher nDCG@5 value signifies more precise and pertinent rankings, where a score of 1 represents the optimal outcome, while 0 reflects the poorest performance.

5.1.1 Evaluation

Based on the evaluation results, we selected the *similar* content with BM25 and *citance* content with SciBERT (these were the best options for both shallow and dense retrieval) to be used in the subsequent summarization step. The former utilizes the top-2 semantically *similar* sentences to the citance (along with the citance itself) as the query with the BM25 model, while the latter employs the *citance* itself as the query with the SciBERT model.

Model	Mean nDCG@5
BM25 (Shallow)	
`similar`	**0.958**
`similar-keywords`	0.944
`citance`	0.943
`neighbors-keywords`	0.928
`citance-keywords`	0.914
`neighbors`	0.898
SciBERT (Dense)	
`citance`	**0.943**
`similar`	0.918
`neighbors`	0.801
`neighbors-keywords`	0.706
`similar-keywords`	0.650
`citance-keywords`	0.617

Table 5.1: Evaluation of the 12 retrieval scenarios (queries based on *citance*, *neighbors*, and *similar*) combined with shallow and dense retrieval models for extracting relevant content from the cited paper. We report nDCG@5 (mean) for 600 relevance judgments. The best model from each retrieval paradigm (in bold) is selected for the summarization step.

5.2 Contextualized Summarization

In this section, we outline the procedure for generating contextualized summaries using extracted contents from the cited paper. Our method utilizes different LLMs (Language Model Models). To begin with, we assessed various instructions and prompt templates to identify the most effective ones for our experiments. Next, we produced contextualized summaries for an evaluation dataset and conducted both manual and automatic evaluations. Below, we will provide further elaboration on the experimental details and the results obtained.

Experimental Details: Using the retrieved content from the cited paper, we employed prompt-based LLMs for abstractive summarization of each citance. We investigated the recently introduced instruction-following models:

1. `Alpaca` (Taori et al., 2023) is finetuned from the LLaMA 7B model (Touvron et al., 2023) using 52K self-instructed instruction-following examples (Wang et al., 2022).

2. `Vicuna` (Chiang et al., 2023) is finetuned from LLaMA using user-shared conversations collected from ShareGPT.[1] It has shown competitive performance when evaluated using `GPT-4` as a judge. We used the 13B variant.

3. `LLaMA-CoT`[2] is a finetuned model on datasets inducing chain-of-thought and logical deductions (Qingyi Si, 2023).

4. `Falcon` (Almazrouei et al., 2023) is trained on the RefinedWeb dataset (Penedo et al., 2023), which is derived through extensive filtering and deduplication of publicly available web data. It is state-of-the-art (at the time of writing) on the open-LLM-leaderboard.[3] We used the 40B-Instruct variant.

5. `GPT-4` (Bubeck et al., 2023) is the latest version of the popular GPT class of models by OpenAI that demonstrates state-of-the-art performance across multiple benchmarks. Therefore, we used it to bootstrap reference summaries for our summarization evaluation (Section 5.2.2). In contrast to the other open-source models, it is accessible exclusively through the OpenAI API.[4]

[1] `https://sharegpt.com/`
[2] `https://huggingface.co/ausboss/llama-30b-supercot`
[3] `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`
[4] `https://platform.openai.com/docs/models/gpt-4`

For the two granularities of retrieved content (top-5 sentences and top-2 paragraphs), we generated separate summaries using the models in a zero-shot setting. For the top-5 sentences, we paraphrased them into coherent text, considering they already served as an extractive summary. For the top-2 paragraphs, we performed abstractive summarization. Throughout the tasks, we experimented with different instructions and prompt formulations. We set the temperature of the models to 0, which gives us deterministic output based only on the input of the relevant content.

5.2.1 Prompt Formulation

To generate a summary conditioned on specific instructions, the mentioned models require clear instructions provided by the user within the prompt. We conducted experiments using various instructions and prompt templates for paraphrasing and summarization.

We used the following instructions for summarization:

1. Generate a coherent summary for the following scientific text in not more than 5 sentences.
2. Generate a short summary of the following scientific text. The summary should not be more than 5 sentences long.
3. Summarize the following scientific text in not more than 5 sentences.

The first two prompts produced concise and understandable summaries. Nevertheless, when the term "coherent" was omitted, the output occasionally appeared as a numbered list comprising separate sentences from the summary. Conversely, the third prompt yielded a highly condensed output, leading to the loss of essential information. To maintain comparability with the top-5 experiment, we restricted the length of the summary for the top-2 experiment to a maximum of 5 sentences.

Instructions for Paraphrasing:

1. Generate a coherent paraphrased text for the following scientific text.
2. Generate a paraphrased text for the following scientific text.
3. Paraphrase the following scientific text.
4. Combine the following scientific text into a coherent and concise text.

The same phenomenon that occurred with the summarizing instructions was also noticed with the first two paraphrasing instructions. However, the third and fourth instructions were frequently misunderstood, leading to the output consisting solely of the input provided.

Prompt templates:

1. Generate a coherent summary for the following scientific text in not more than 5 sentences.

   scientific text: {*input*}

   summary:

2. A chat between a curious user and an artificial intelligence assistant. The assistant knows how to summarize scientific text and the user will provide the scientific text for the assistant to summarize.

   USER:
   Generate a coherent summary for the following scientific text in not more than 5 sentences: {*input*}.

   ASSISTANT:

3. ### **Instruction:**
   A chat between a curious human and an artificial intelligence assistant. The assistant knows how to paraphrase scientific text and the user will provide the scientific text for the assistant to paraphrase.

   ### **Input:**
   Generate a coherent paraphrased text for the following scientific text: {*input*}.

   ### **Output:**

Figure 5.1: Illustration of Prompt Templates for Paraphrasing and Summarization. While Template 1 yielded favorable outcomes for the `Alpaca` and `Falcon` models, it proved ineffective for `Vicuna` and `LLaMA-CoT`. Template 2 demonstrated success with `Alpaca`, `LLaMA-CoT`, and `Falcon`, yet fell short with `Vicuna`. Template 3 produced cohesive summaries across all models, albeit occasionally exhibiting artifacts.

We used a direct instruction prompt without a template for `GPT-4`. The first template worked well for the `Alpaca` and `Falcon` models but proved inadequate for the `Vicuna` and `LLaMA-CoT` models. When the input's last sentence had a colon or an abbreviation, the model tried to align the output with it. Specifically, the `Vicuna` model represented summaries as bullet points, but the `LLaMA-CoT` model failed to generate any output using this template, rendering it unusable. The second template worked well for `Alpaca`, `LLaMA-CoT`, and `Falcon`. Unfortunately, it did not generate any output for `Vicuna`. Eventually, we opted for the third prompt template, which successfully generated readable, coherent, and understandable summaries for all four models. However, it did occasionally result in artifacts in some individual examples, though it was the most effective overall.

To determine the best combination of instructions and prompt formulations for each model, we manually evaluated summaries generated from ten examples for each model. Based on this evaluation, we selected the optimal combination, as depicted in Figure 5.2, to use for the final evaluation.

Paraphrasing Prompt

Instruction:
A chat between a curious human and an artificial intelligence assistant. The assistant knows how to paraphrase scientific text and the user will provide the scientific text for the assistant to paraphrase.
Input:
Generate a coherent paraphrased text for the following scientific text: {*input*}.
Output:

Summarization Prompt

Instruction:
A chat between a curious human and an artificial intelligence assistant. The assistant knows how to summarize scientific text and the user will provide the scientific text for the assistant to summarize.
Input:
Generate a coherent summary for the following scientific text in not more than 5 sentences: {*input*}.
Output:

Figure 5.2: Best prompts with instructions used for paraphrasing (of top-5 sentences) and summarization (of top-2 paragraphs). We ensured similar summary lengths for both granularities by strictly instructing the model to generate not more than 5 sentences for the top-2 paragraphs.

5.2.2 Evaluation

We assessed the contextualized summaries produced by the models mentioned in Section 5.2. We developed an evaluation dataset and created ground truth summaries with `GPT-4` to evaluate them. We used ROUGE and BertScore metrics for quantitative evaluation and had three experts assess our summaries for coverage and focus for qualitative evaluation. Below, we describe their agreement and the **results** obtained, along with examples.

ROUGE

Our evaluation involved the use of ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004) metrics to automatically score the summaries generated by `Alpaca`, `Falcon`, `LLaMA-CoT`, and `Vicuna` models, compared to the reference summary by `GPT-4`. ROUGE-1 and ROUGE-2 measured the agreement of one-word and two-word combinations, respectively, while ROUGE-L computed the agreement of the longest common subsequence. This determined the number of matching tokens in the correct order, even if not necessarily consecutive. The more tokens overlap, the higher the value of these metrics, indicating the similarity between the generated summaries and reference summaries.

BertScore

We also used BertScore (Zhang et al., 2020), an automatic metric for measuring the quality of generated texts. This metric measures the similarity score of each token in the generated summary to those in the reference summary. Unlike exact matches, BertScore employs contextual embeddings. It first generates BERT embeddings for each token in both the generated and reference summary. Then it computes pairwise cosine similarity and selects the highest cosine similarity values using a greedy matching method. Finally, BertScore is calculated by weighting the cosine similarity values with inverse document frequency (IDF) weights, summing them, and dividing them by the sum of the IDF weights.

Evaluation Data

We selected 15 papers from the ACL anthology published between 2016-2020, aligned with the NLP domain. We extracted 363 citances from these papers and randomly chose 25 of them. Using the full texts of the cited papers and the two best retrieval models from Table 5.1, we retrieved the top-5 sentences and top-2 paragraphs, resulting in 100 texts in total. To create the ground-truth summaries, we utilized `GPT-4` (Bubeck et al., 2023) in a zero-shot setting to

paraphrase/summarize these texts. Each summary was manually verified to ensure accuracy and eliminate hallucinations or factual errors. We employed the prompts shown in Figure 5.2.

Automatic Evaluation

We used the ground-truth summaries to evaluate the contextualized summaries from the LLMs automatically. Specifically, we computed ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020) using the evaluation module of the summary workbench (Syed et al., 2022). Results shown in Table 5.2 reveal that, according to ROUGE, `Vicuna` performs best for (summarizing) the top-2 paragraphs, while `LLaMA-CoT` is the best for (paraphrasing) the top-5 sentences as the summary. Moreover, it also achieves the highest BERTScore in the top-2 paragraphs setting. Accordingly, we manually evaluated them for coverage and focus.

Human Evaluation

We recruited three annotators from the NLP domain, including two Ph.D. students and one post-doc, to assess the usefulness of the summaries. The annotators were requested to rate the summaries on two criteria: *coverage* and *focus*. The ratings were on a scale of 1 to 5, with 1 representing the worst and 5 representing the best. *Coverage* reflects how well the summary captures the essential information from the cited paper that is relevant to a specific citance, while *focus* pertains to the coherence and cohesion of the sentences in the summary. A total of 125 summaries were evaluated for papers cited from 25 unique citances. Each example consisted of the citance (and its context) displayed on the left side, accompanied by five summaries on the right: the abstract of the cited paper, two reference summaries (top-5 sentences and top-2 paragraphs), and the summaries generated by the two best models for the top-5 sentences and top-2 paragraphs scenarios, namely `Vicuna-similar-BM25-top2` and `LLaMA-CoT-citance-SciBert-top5`. The order of the summaries was randomized to mitigate any sequence effects (Mathur et al., 2017). Table 5.4 shows one example that the human annotators evaluated. In the evaluation scenario, the model names and retrieval scenarios were masked.

Model	BERTScore	ROUGE		
		R-1	R-2	R-L
	top-2 paras.			
similar-bm25				
Alpaca	0.343	47.3	25.5	44.9
Falcon	0.401	48.2	27.1	45.0
LLaMA-CoT	0.448	53.0	31.9	50.5
Vicuna	0.465	**58.7**	**35.4**	**55.8**
citance-SciBERT				
Alpaca	0.390	54.3	32.2	52.0
Falcon	0.413	52.1	29.6	48.9
LLaMA-CoT	**0.497**	54.7	32.9	52.5
Vicuna	0.431	56.7	34.2	53.9
	top-5 sents.			
similar-bm25				
Alpaca	0.616	56.2	35.4	54.8
Falcon	0.649	57.5	35.6	55.2
LLaMA-CoT	0.707	61.2	38.6	60.0
Vicuna	0.551	57.2	34.3	54.9
citance-SciBERT				
Alpaca	0.595	56.6	34.7	55.1
Falcon	0.656	56.8	36.2	55.3
LLaMA-CoT	**0.748**	**62.9**	**40.6**	**60.9**
Vicuna	0.607	58.8	36.0	56.6

Table 5.2: Automatic evaluation of summaries from all LLMs grouped by two granularities: top-2 relevant paragraphs and top-5 relevant sentences from the cited paper. We report BERTScore (precision) and ROUGE scores against the reference summaries from GPT-4. We chose the best model from each scenario based on ROUGE overlap with the references for manual evaluation: Vicuna (*similar-BM25*) and LLaMA-CoT (*citance-SciBERT*) for top-2 paragraphs and top-5 sentences, respectively.

Inter Annotator Agreement and Results

We computed inter-annotator agreement using weighted Cohen's Kappa (Cohen, 1960) for our ordinal data and obtained κ of 0.42 and 0.40 for coverage and focus, respectively. While these results indicate a fair to moderate agreement among the annotators (Appendix C.1), we recognize that the evaluation task itself is subjective. Assessing the usefulness of a summary is influenced by various contextual factors, such as the annotators' goals when reviewing a citation, their prior knowledge about the cited paper, and the presentation of the

summary (Jones, 2007). Overall, the abstracts received slightly higher scores from the annotators in terms of both coverage and focus. As shown in Table 5.3, the abstract achieved the highest coverage score (3.67), closely followed by the `Vicuna` summary (3.01). Similarly, for focus, the abstract was rated as the best summary (4.50), with the reference summary from `GPT-4` coming in second (3.83). Regarding the granularity of the retrieved content, we observed that summarizing the top-2 paragraphs outperformed summarizing the top-5 sentences in terms of both coverage and focus.

Model	**Coverage**	**Focus**
`Abstract`	**3.67**	**4.50**
similar-bm25, top-2 paras		
`Reference (GPT4)`	2.92	3.83
`Vicuna`	3.01	3.56
citance-SciBERT, top-5 sent.		
`Reference (GPT4)`	2.45	2.99
`LLaMA-CoT`	2.33	2.33

Table 5.3: Average scores for summary quality criteria (over 125 summaries) as per human evaluation. Models are grouped by the retrieval scenario.

5.2.3 Summary Examples

Despite the general preference for abstracts, feedback from annotators indicated that our summaries were more effective when a citance did not directly reference the main contributions of a paper. The abstract was preferred over the contextualized summaries for the Example 5.4. The citance in this example reads, *"The skill implements Hybrid Code Networks (HCNs) described in (Williams et al., 2017)"*. It cites Hybrid Code Networks, which are the main contributions of the publication *"Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning"* (Williams et al., 2017). Therefore the essential information about Hybrid Code Networks is included in the abstract.

DeepPavlov: Open-Source Library for Dialogue Systems
`Citance Context`
Some of them are available for interactive online testing.
The skill implements Hybrid Code Networks (HCNs) described in (Williams et al., 2017).
The model is configurable: embeddings, slot filling component and intent classifier can be switched on and off on demand.
`Abstract`
End-to-end learning of recurrent neural networks (RNNs) is an attractive solution for dialog systems; however, current techniques are data-intensive and require thousands of dialogs to learn simple behaviors. We introduce Hybrid Code Networks (HCNs), which combine an RNN with domain-specific knowledge encoded as software and system action templates. Compared to existing end-to-end approaches, HCNs considerably reduce the amount of training data required, while retaining the key benefit of inferring a latent representation of dialog state. In addition, HCNs can be optimized with supervised learning, reinforcement learning, or a mixture of both. HCNs attain state-of-the-art performance on the bAbI dialog dataset, and outperform two commercially deployed customer-facing dialog systems.
`GPT-citance-SciBert-top5`
After every dialogue, RL policy updates occur. State is maintained by both the RNN and the developer code. HCNs employ an RNN to gather dialogue state and select actions as well. In these structures, the RNN learns to handle entity values, such as storing them in memory. On the other hand, HCNs utilize an RNN that inherently deduces a state representation.
`GPT-similar-BM25-top2`
This paper introduces Hybrid Code Networks (HCNs), a model for end-to-end learning that combines learning an RNN with domain knowledge expressed via software and action templates. HCNs achieve the same performance as existing recurrent end-to-end techniques but with significantly less training data. The neural network can be trained with supervised learning or reinforcement learning by changing the gradient update applied. The model is applied to the bAbI dialog dataset and real customer support domains. HCNs can be optimized with reinforcement learning, using LSTM for the recurrent layer and the AdaDelta optimizer.
`LLaMA-CoT-citance-SciBert-top5`
The RL policy is updated after each dialogue. Both the RNN and the developer code retain state. HCNs also use an RNN to accumulate dialogue state and select actions. In each of these architectures, the RNN learns to manipulate entity values, for example by saving them in a memory. By contrast, HCNs use an RNN which automatically infers a representation of state.
`Vicuna-similar-BM25-top2`
This paper presents a new model called Hybrid Code Networks (HCNs) for end-to-end learning. HCNs allow developers to express domain knowledge and achieve better performance with less training data compared to existing techniques. The model is trained using supervised or reinforcement learning and can be applied to the bAbI dialog dataset and real customer support domains. The model can also be optimized with reinforcement learning.

Table 5.4: Example of the automatically generated contextualized summaries from the best models for both granularities. In this example, the citance cites the main contribution of the cited paper.

On the other hand, contextualized summaries were favored when the main contributions were not cited, as seen in the example presented in Table 5.5. In this second example, the citance reads, *"In both cases, we use the BIOES labeling scheme for the output tags, following previous work which showed it outperforms other options (e.g., Ratinov and Roth, 2009)."*. However, this citance does not refer to the main contribution of the paper *"Design Challenges and Misconceptions in Named Entity Recognition"* (Ratinov and Roth, 2009). It references a BIOES labeling scheme that lacks any information within the abstract. Therefore, in this example, the contextualized summaries produced by `GPT-similar-BM25-top2` and `Vicuna-similar-BM25-top2` were rated higher than the abstract.

Semi-supervised sequence tagging with bidirectional language models

`Citance Context`

In both cases, we use the BIOES labeling scheme for the output tags, following previous work which showed it outperforms other options (e.g., Ratinov and Roth, 2009).

In addition to CNN-BIG-LSTM from Józefowicz et al. (2016), 1 we used the same corpus to train two additional language models with fewer parameters: forward LSTM-2048-512 and backward LSTM-2048-512. In all experiments, we fine tune the pre-trained Senna word embeddings but fix all weights in the pre-trained language models.

`Abstract`

We analyze some of the fundamental design challenges and misconceptions that underlie the development of an efficient and robust NER system. In particular, we address issues such as the representation of text chunks, the inference approach needed to combine local NER decisions, the sources of prior knowledge and how to use them within an NER system. In the process of comparing several solutions to these challenges we reach some surprising conclusions, as well as develop an NER system that achieves 90.8 F1 score on the CoNLL-2003 NER shared task, the best reported result for this dataset.

`GPT-citance-SciBert-top5`

In order to achieve the most accurate comparison, we provided each system with unprocessed text and relied on the system's ability to split sentences and tokenize them. This approach divides the Viterbi decision process into separate maximization tasks over smaller segments, where a greedy strategy is effective. Perceptron-based systems have demonstrated competitiveness in Named Entity Recognition (NER) and text chunking (Kazama and Torisawa, 2007b; Punyakanok and Roth, 2001; Carreras et al., 2003). We defined the model and its features using the LBJ (Rizzolo and Roth, 2007) modeling language. It is important to note that in this experiment, we also utilized token-level accuracy on the CoNLL dataset. The BILOU scheme proposes learning classifiers that can identify the Beginning, Inside, and Last tokens of multi-token chunks, as well as Unit-length chunks.

`GPT-similar-BM25-top2` We analyze some of the fundamental design challenges and misconceptions that underlie the development of an efficient and robust NER system. In particular, we address issues such as the representation of text chunks, the inference approach needed to combine local NER decisions, the sources of prior knowledge and how to use them within an NER system. In the process of comparing several solutions to these challenges we reach some surprising conclusions, as well as develop an NER system that achieves 90.8 F1 score on the CoNLL-2003 NER shared task, the best reported result for this dataset.
`LLaMA-CoT-citance-SciBert-top5` For the most accurate comparison, we have presented each system with raw text, and relied on the system's sentence splitter and tokenizer. This separation breaks the Viterbi decision process into independent maximization of assignment over short chunks, where the greedy policy performs well. Systems based on perceptron have been shown to be competitive in NER and text chunking (Kazama and Torisawa, 2007b; Punyakanok and Roth, 2001; Carreras et al., 2003). We specify the model and the features with the LBJ (Rizzolo and Roth, 2007) modeling language. Also note that in this experiment we have used token-level accuracy on the CoNLL dataset as well. The BILOU scheme suggests to learn classifiers that identify the Beginning, the Inside and the Last tokens of multi-token chunks as well as Unit-length chunks.
`Vicuna-similar-BM25-top2` A simple technique of using word clusters generated from unlabeled text can improve performance of dependency parsing, Chinese word segmentation, and NER. The technique is based on word class models and uses a binary tree to represent words. The approach is related to distributional similarity, but not identical. The system's performance is significantly impacted by the choice of encoding scheme, and the less used BILOU formalism outperforms the widely adopted BIO tagging scheme.

Table 5.5: Example of the automatically generated contextualized summaries from the best models for both granularities. In this example, the citance doesn't cite the main contribution of the cited paper.

Chapter 6

Discussion & Conclusion

In the following section, we aim to delve into the limitations encountered during our extensive experiments and evaluations. We will list these limitations regarding the dataset, the relevant content retrieval, and contextualized summarization. Additionally, we will conclude by providing an overall summary of the thesis, highlighting the most significant contributions, and outlining potential future research questions and approaches for improving the process of generating and evaluating contextualized summaries. I want to acknowledge that computations for this work were done (in part) using the resources of the Leipzig University Computing Center[1].

Dataset Limitations

The developed CONTEXT-SCISUMM corpus, while extensive and of high quality, is currently confined to the field of computer science. Exploring how contextualized summaries perform in other domains would be interesting. However, care must be taken, especially in disciplines such as medicine, where vital information could be lost through the summarization process, e.g., specific drug dosage information, potentially leading to dangerous consequences.

Another aspect to consider is the variation in citation styles and practices across different scientific disciplines. Before expanding the dataset, a thorough analysis of the citation style and domain is imperative. This ensures that unique characteristics are acknowledged and utilized, enabling the extraction of more precise contextual information from the cited paper and consequently generating more accurate contextualized summaries.

Regarding the evaluation dataset (Section 5.2.2), it is a subset of CONTEXT-SCISUMM and includes not only computer science papers but also only 15 papers from the NLP domain. Expanding this dataset to encompass broader

[1] https://www.sc.uni-leipzig.de/

areas of computer science and diverse disciplines would be advisable. To accomplish this endeavor, it would be necessary to bring in additional experts from various areas of computer science as well as other scientific disciplines to expand the dataset, which is particularly important given the moderate agreement among the three reviewers for this evaluation dataset. This action is critical to ensure the quality of the contextualized summaries across various disciplines.

Retrieval Limitations

Our analysis of both shallow and dense retrieval models has highlighted the necessity for improved relevance judgments achieved through consensus among multiple annotators and a broader selection of samples. A single individual performed relevance judgments for ten queries in 12 different retrieval scenarios, yielding 600 sample instances. Furthermore, we exclusively performed relevance assessments for the top-5 sentence approach, neglecting the paragraph approach. This might have led to a suboptimal selection of retrieval scenarios for the top-2 paragraphs. As a result, it is imperative to increase the number of relevance judgments from multiple annotators and include relevance judgments for the top-2 paragraphs to obtain meaningful results for each retrieval scenario.

Regarding modeling the citation-context, in the *similar* approach, we employed two semantically similar sentences in a paragraph to the citance without implementing a predetermined threshold for the cosine similarity. In the *neighbors* approach, we included surrounding sentences without further filtering. The absence of filtering or a threshold could potentially lead to the inclusion of irrelevant information, thereby adversely affecting the precision of the query. Furthermore, there could be the introduction of noise when a citance contains multiple references to papers from distinct subjects. Therefore, it is imperative to disentangle individual citations within a citance (Schwartz and Hearst, 2006), allowing for the specific modeling of citation-context for each individual citance. This, in turn, enhances the accuracy of the query for retrieving the pertinent context.

Furthermore, we retrieved the top-5 sentences and top-2 paragraphs from the cited paper as the relevant content for a citance. Employing fixed numbers could lead to excluding some appropriate sentences and paragraphs or including inappropriate sentences and paragraphs. Implementing a dynamic variant with thresholds to decide which sentences to include or exclude would be more appropriate. The length had to be limited due to limited access to graphics cards. We had access to four Nvidia V100s. Additionally, the cost of using `GPT-4` increases significantly as the extracted content grows.

Summarization Challenges

In the current framework for generating contextualized summaries, we must address several limitations inherent in our current approach. Key areas demanding improvement encompass our instructions and prompt templates, the utilization of LLMs, our evaluation framework, and the metrics deployed to gauge summary quality. By dissecting these aspects and implementing requisite adjustments, we can work towards ensuring that our summarization process is as effective and reliable as possible.

We identified a particular pairing of instruction and prompt template that yielded the most favorable outcomes for summarization and paraphrasing while also aiming to reduce inaccurate outputs. Despite our efforts, we did come across instances of flawed outputs that had an adverse impact on evaluation scores (Appendix B.1). As a result, additional efforts are required to enhance and fine-tune the prompt templates and instructions, with the goal of further diminishing or eliminating these erroneous outputs.

In addition, our evaluation approach may be extended to more directly account for direct informativeness. We claim that contextualized summaries provide higher information gain than abstracts, especially when a paper is cited multiple times in the same citing paper, but this case was not adequately considered in our evaluation and warrants further analysis. Conversations with the evaluators noted that they preferred contextualized summaries when the citance did not reference the main idea of a scientific paper, which the abstract usually covers. Despite masking, they sometimes identified the abstract among the five examples and still felt that the contextualized summaries were superior in those cases. Confirming these subjective statements would necessitate a more substantial evaluation.

Furthermore, our approach relies on LLMs, which are constantly evolving and being investigated by the research community. It is worth noting that the results of our experiments may vary with the introduction of newer LLMs. However, the underlying approach is intuitive and can be easily adapted to incorporate newer LLMs as they become available. It is crucial to note that the size of LLMs and the accessibility of graphics cards are restricted, creating a further constraint that must be considered.

It is also important to recognize a significant yet often overlooked limitation of any summarization approach, which is the lack of a clear definition of what constitutes a good summary, considering the purpose of the summary. In our case, the purpose of the summary is to assist readers in understanding a citance without having to refer to the cited paper. While we used the abstract as a reference point for comparison, our evaluation methodology lacks a concrete integration of the summary's purpose. This makes a fair comparison between

the abstract and the contextualized summary challenging. Additionally, our evaluation was primarily focused on the NLP domain due to the availability of expert annotators, which means that the results may not be directly applicable to scientific papers from other domains. We hope this work will inspire the research community to develop a more robust evaluation methodology for contextualized summarization that aligns with the specific purposes of different summaries.

6.1 Conclusion & Future Work

In this thesis, we focused on creating abstractive, informative, and contextualized summaries for scientific papers, aiming to enhance the understanding of a scientific paper and expedite the reading process. While abstracts exist for scientific papers, they often lack comprehensiveness in providing sufficient information about the content of a paper. Therefore, our work centers on developing tailored contextualized summaries. The three main contributions of this research are as follows:

1. A large-scale, high-quality dataset for contextualized summarization of scientific papers
2. A framework to generate abstractive, informative, and contextualized summaries given a citance
3. Qualitative and quantitative evaluation of the corpus, relevant content extraction, and the contextualized summaries

First and foremost, we constructed a comprehensive and high-quality dataset CONTEXT-SCISUMM specifically designed for contextualized summaries of scientific papers, comprising 540K papers and 4.6M citances from the computer science domain. This dataset was generated using our novel framework and is based on the S2ORC dataset.

Subsequently, we devised a framework for creating contextualized summaries encompassing three crucial steps. In the first step, we modeled the citation-context, exploring different types of implicit citation-contexts to enhance the relevance and accuracy of the query. The second step focused on citance-guided information retrieval using both shallow and dense retrieval models at the sentence and paragraph levels. Manual assessments were employed to determine the most effective models. Finally, the third step utilized state-of-the-art LLMs like `GPT-4`, `Falcon`, `LLaMa-Cot`, `Vicuna`, and `Alpaca` to generate contextualized summaries. This process involved paraphrasing the

top-5 most relevant sentences and summarizing the top-2 most significant paragraphs. We experimented with different prompt templates and instructions.

In the third facet of our research, we conducted qualitative and quantitative evaluations of the corpus, relevant content extraction approaches, and contextualized summaries. We compared our contextualized summaries with traditional abstracts to gauge their effectiveness in aiding comprehension of citances and, therefore, the citing paper. Our experiments with zero-shot summarization using LLMs revealed that abstracts are slightly preferred over contextualized summaries regarding coverage and focus. However, contextualized summaries performed better when citances did not reference the general idea of a paper. We also demonstrated the robustness of our approach regarding the choice of LLMs, with `GPT-4` showing the best results.

Apart from the issues mentioned in the discussion section, there are other aspects where this work can be enhanced and expanded in the future. Initially, addressing the challenge of handling multiple citations within a single citance is crucial. Dealing with multiple citations can result in inaccuracies in modeling the citation-context, impacting the extraction of relevant content. A more detailed exploration of the structure and information within such citances is necessary to overcome this obstacle.

An exciting possibility for extension involves using LLMs like `GPT-4` to retrieve the relevant content and model the citation-context. This approach would entail inputting the full text of the cited paper and the citance, with specifically tailored prompts to extract the relevant content. A similar strategy could be used for modeling the citation-context, with the citance and the surrounding paragraph as input. However, there may be challenges related to the content size during the LLM-retrieval process, as well as constraints imposed by the models or GPUs. Additionally, using `GPT-4` would come with higher costs.

Another area for improvement is related to evaluations, including relevance assessment and human evaluation of summaries. It is essential to include more evaluators covering different disciplines beyond computer science to evaluate a broader range of citances. Since metrics used in human evaluation are subjective, it is crucial to accurately define the evaluation criteria and carefully examine the factors considered. In addition to a more comprehensive and precise evaluation, we intend to use G-Eval (Liu et al., 2023) for prompt-based evaluation. Specially designed prompts will enable evaluations across different disciplines without the need for numerous experts. A series of future experiments will be conducted to assess how well these evaluations align with experts' assessments. If a high level of agreement is observed, a large-scale evaluation could be performed to gain deeper insights into creating citance-contextualized

summaries and further optimize scientific papers' comprehension.

In conclusion, our research makes a notable contribution to the field of summarization by providing a valuable framework for generating citance-contextualized summaries of scientific papers. The generated summaries have the potential to significantly improve the understanding of citances and facilitate the reading process for researchers and scientists. However, we acknowledge that our approach currently exhibits certain weaknesses and that numerous opportunities for refinement and enhancement still exist.

Bibliography

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 500–509, Portland, Oregon, USA. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Nouf Ibrahim Altmami and Mohamed El Bachir Menai. 2022. Automatic summarization of scientific articles: A survey. *J. King Saud Univ. Comput. Inf. Sci.*, 34(4):1011–1028.

Myriam Hernández Álvarez and José M. Gómez. 2016. Survey about citation context analysis: Tasks, techniques, and resources. *Nat. Lang. Eng.*, 22(3):327–349.

Phyllis B. Baxendale. 1958. Machine-made index for technical literature - an experiment. *IBM J. Res. Dev.*, 2(4):354–361.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa

Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Dorian Brown. 2020. Rank-BM25: A Collection of BM25 Algorithms in Python.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S. Weld. 2020. TLDR: extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4766–4777. Association for Computational Linguistics.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Eduard Hovy, Philipp Mayr, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview of the first workshop on scholarly document processing (SDP). In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 1–6, Online. Association for Computational Linguistics.

Yifan Chen, Tamara Polajnar, Colin Batchelor, and Simone Teufel. 2020. A corpus of very short scientific summaries. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 153–164, Online. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 390–400, Lisbon, Portugal. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2017. Contextualizing citations for scientific summarization using word embeddings and domain knowledge. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1133–1136. ACM.

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *Int. J. Digit. Libr.*, 19(2-3):287–303.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. 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. Association for Computational Linguistics.

Danish Contractor, Yufan Guo, and Anna Korhonen. 2012. Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012*, pages 663–678, Mumbai, India. The COLING 2012 Organizing Committee.

Aaron Elkiss, Siwei Shen, Anthony Fader, Günes Erkan, David J. States, and Dragomir R. Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *J. Assoc. Inf. Sci. Technol.*, 59(1):51–62.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin,

Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Francesca Bonin, and David Konopnicki. 2019. 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. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 160–168. AAAI Press.

Vivek Gupta, Prerna Bharti, Pegah Nokhiz, and Harish Karnick. 2021. SumPubMed: Summarization dataset of PubMed scientific articles. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 292–303, Online. Association for Computational Linguistics.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Karen Spärck Jones. 2007. Automatic summarising: The state of the art. *Inf. Process. Manag.*, 43(6):1449–1481.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2125–2131. Association for Computational Linguistics.

Maria Liakata, Simon Dobnik, Shyamasree Saha, Colin Batchelor, and Dietrich Rebholz-Schuhmann. 2013. A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 747–757, Seattle, Washington, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using GPT-4 with better human alignment. *CoRR*, abs/2303.16634.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, Raymond Fok, Fangzhou Hu, Regan Huff, Dongyeop Kang, Tae Soo Kim, Rodney Kinney, Aniket Kittur, Hyeonsu B. Kang, Egor Klevak, Bailey Kuehl, Michael Langan, Matt Latzke, Jaron Lochner, Kelsey MacMillan, Eric Marsh, Tyler Murray, Aakanksha Naik, Ngoc-Uyen Nguyen, Srishti Palani, Soya Park, Caroline Paulic, Napol Rachatasumrit, Smita Rao, Paul Sayre, Zejiang Shen, Pao Siangliulue, Luca Soldaini, Huy Tran, Madeleine van Zuylen, Lucy Lu Wang, Chris Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Marti A. Hearst, and Daniel S. Weld. 2023. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *CoRR*, abs/2303.14334.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165.

Yuning Mao, Ming Zhong, and Jiawei Han. 2022. CiteSum: Citation text-guided scientific extreme summarization and domain adaptation with limited supervision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10922–10935, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2017. Sequence effects in crowdsourced annotations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2860–2865. Association for Computational Linguistics.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of ACL-08: HLT*, pages 816–824, Columbus, Ohio. Association for Computational Linguistics.

Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. 2021. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1080–1089, Online. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: outperforming curated corpora with web data, and web data only. *CoRR*, abs/2306.01116.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696, Manchester, UK. Coling 2008 Organizing Committee.

Vahed Qazvinian and Dragomir R. Radev. 2010. Identifying non-explicit citing sentences for citation-based summarization. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 555–564. The Association for Computer Linguistics.

Siya Qi, Lei Li, Yiyang Li, Jin Jiang, Dingxin Hu, Yuze Li, Yingqi Zhu, Yanquan Zhou, Marina Litvak, and Natalia Vanetik. 2022. SAPGraph: Structure-aware extractive summarization for scientific papers with heterogeneous graph. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1:*

Long Papers), pages 575–586, Online only. Association for Computational Linguistics.

Zheng Lin Qingyi Si. 2023. Alpaca-cot: An instruction fine-tuning platform with instruction data collection and unified large language models interface. `https://github.com/PhoebusSi/alpaca-CoT`.

Napol Rachatasumrit, Jonathan Bragg, Amy X. Zhang, and Daniel S. Weld. 2022. Citeread: Integrating localized citation contexts into scientific paper reading. In *IUI 2022: 27th International Conference on Intelligent User Interfaces, Helsinki, Finland, March 22 - 25, 2022*, pages 707–719. ACM.

Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The ACL anthology network corpus. *Lang. Resour. Evaluation*, 47(4):919–944.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*, pages 147–155. ACL.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Agata Rotondi, Angelo Di Iorio, and Freddy Limpens. 2018. Identifying citation contexts: a review of strategies and goals. In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Torino, Italy, December 10-12, 2018*, volume 2253 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Tim Schopf, Simon Klimek, and Florian Matthes. 2022. Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2022, Volume 1: KDIR, Valletta, Malta, October 24-26, 2022*, pages 243–248. SCITEPRESS.

Ariel S. Schwartz and Marti A. Hearst. 2006. Summarizing key concepts using citation sentences. In *Proceedings of the Workshop on Linking Natural Language and Biology, BioNLP@NAACL-HLT 2006, New York, NY, USA, June 8, 2006*, pages 134–135. Association for Computational Linguistics.

Amir Soleimani, Vassilina Nikoulina, Benoit Favre, and Salah Ait Mokhtar. 2022. Zero-shot aspect-based scientific document summarization using self-supervised pre-training. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 49–62, Dublin, Ireland. Association for Computational Linguistics.

Sajad Sotudeh, Arman Cohan, and Nazli Goharian. 2021. On generating extended summaries of long documents. In *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Sajad Sotudeh and Nazli Goharian. 2022. TSTR: Too short to represent, summarize with details! intro-guided extended summary generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–335, Seattle, United States. Association for Computational Linguistics.

Shahbaz Syed, Dominik Schwabe, and Martin Potthast. 2022. SUMMARY WORKBENCH: unifying application and evaluation of text summarization models. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022 - System Demonstrations, Abu Dhabi, UAE, December 7-11, 2022*, pages 232–241. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Simone Teufel and Marc Moens. 2002. Articles summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Juan-Manuel Torres-Moreno. 2014. *Automatic Text Summarization*. Wiley.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *CoRR*, abs/2212.10560.

Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 665–677. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7386–7393. AAAI Press.

Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, Naif Radi Aljohani, and Raheel Nawaz. 2020. HTSS: A novel hybrid text summarisation and simplification architecture. *Inf. Process. Manag.*, 57(6):102351.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Appendix A

Relevant Content Files

Top-2 Paragraphs:

Listing A.1: The top-2 paragraphs were selected using the retrieval scenarios, *similar-BM25*, *similar-keywords-BM25*, *citance-SciBERT*, and *citance-BM25*. The paragraphs are extracted from the paper *Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning* (Williams et al., 2017)


```
[
   {
  "citance_No": 5,
  "citing_paper_id": 51871042,
  "similar-BM25": "This paper presents a model for end
     -to-end learning, called Hybrid Code Networks (
     HCNs) which addresses these problems. In addition
      to learning an RNN, HCNs also allow a developer
     to express domain knowledge via software and
     action templates. Experiments show that, compared
      to existing recurrent end-to-end techniques,
     HCNs achieve the same performance with
     considerably less training data, while retaining
     the key benefit of end-to-end trainability.
     Moreover, the neural network can be trained with
     supervised learning or reinforcement learning, by
      changing the gradient update applied. This paper
      is organized as follows. Section 2 describes the
      model, and Section 3 compares the model to
     related work. Section 4 applies HCNs to the bAbI
     dialog dataset (Bordes and Weston, 2016) .
     Section 5 then applies the method to real
     customer support domains at our company. Section
```

6 illustrates how HCNs can be optimized with reinforcement learning, and Section 7 concludes. We then trained an HCN on the training set, employing the domain-specific software described above. We selected an LSTM for the recurrent layer (Hochreiter and Schmidhuber, 1997) , with the AdaDelta optimizer (Zeiler, 2012). We used the development set to tune the number of hidden units (128), and the number of epochs (12). Utterance embeddings were formed by averaging word embeddings, using a publicly available 300 dimensional word embedding model trained using word2vec on web data (Mikolov et al., 2013). 4 The word embeddings were static and not updated during LSTM training. In training, each dialog formed one minibatch, and updates were done on full rollouts (i.e., non-truncated back propagation through time). The training loss was categorical cross-entropy. Further low-level implementation details are in the Appendix Section A.1.",
"similar-keywords-BM25": "We compare to four past end-to-end approaches (Bordes and Weston, 2016; Liu and Perez, 2016; Eric and Manning, 2017; Seo et al., 2016) . We emphasize that past approaches have applied purely sequence-to-sequence models, or (as a baseline) purely programmed rules (Bordes and Weston, 2016) . By contrast, Hybrid Code Networks are a hybrid of hand-coded rules and learned models. At a high level, the four components of a Hybrid Code Network are a recurrent neural network; domain-specific software; domain-specific action templates; and a conventional entity extraction module for identifying entity mentions in text. Both the RNN and the developer code maintain state. Each action template can be a textual communicative action or an API call. The HCN model is summarized in Figure 1 .",
"citance-SciBERT": "For optimization, we selected a policy gradient approach (Williams, 1992) , which has been successfully applied to dialog systems (Jur\\u010d\\u00ed\\u010dek et al., 2011) , robotics (Kohl and Stone, 2004) , and the board

game Go (Silver et al., 2016) . In policy gradient-based RL, a model \\u03c0 is parameterized by w and outputs a distribution from which actions are sampled at each timestep. At the end of a dialog, the return G for that dialog is computed, and the gradients of the probabilities of the actions taken with respect to the model weights are computed. The weights are then adjusted by taking a gradient step proportional to the return: In summary, HCNs can out-perform production-grade rule-based systems with a reasonable number of labeled dialogs, and adding synthetic \\sunny-day\\ dialogs improves performance further. Moreover, unlike existing pipelined approaches to dialog management that rely on an explicit state tracker, this HCN used no explicit state tracker, highlighting an advantage of the model.",
"citance-BM25": "We compare to four past end-to-end approaches (Bordes and Weston, 2016; Liu and Perez, 2016; Eric and Manning, 2017; Seo et al., 2016) . We emphasize that past approaches have applied purely sequence-to-sequence models, or (as a baseline) purely programmed rules (Bordes and Weston, 2016) . By contrast, Hybrid Code Networks are a hybrid of hand-coded rules and learned models. This paper presents a model for end-to-end learning, called Hybrid Code Networks (HCNs) which addresses these problems. In addition to learning an RNN, HCNs also allow a developer to express domain knowledge via software and action templates. Experiments show that, compared to existing recurrent end-to-end techniques, HCNs achieve the same performance with considerably less training data, while retaining the key benefit of end-to-end trainability. Moreover, the neural network can be trained with supervised learning or reinforcement learning, by changing the gradient update applied. This paper is organized as follows. Section 2 describes the model, and Section 3 compares the model to related work. Section 4 applies HCNs to the bAbI dialog dataset (Bordes and Weston, 2016) . Section 5 then

```
        applies the method to real customer support
        domains at our company. Section 6 illustrates how
         HCNs can be optimized with reinforcement
        learning, and Section 7 concludes. "
    }
   ]
```


Top-5 Sentences:

Listing A.2: The top-5 sentences were selected using the retrieval scenarios, *similar-BM25*, *similar-keywords-BM25*, *citance-SciBERT*, and *citance-BM25*. The sentences are extracted from the paper *Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning* (Williams et al., 2017)


```
[
   {
  "citance_No": 5,
  "citing_paper_id": 51871042,
  "similar-BM25": "This paper presents a model for end-to-
     end learning, called Hybrid Code Networks (HCNs)
     which addresses these problems. By contrast, Hybrid
     Code Networks are a hybrid of hand-coded rules and
     learned models. Utterance embeddings were formed by
     averaging word embeddings, using a publicly available
      300dimensional word embedding model trained using
     word2vec on web data (Mikolov et al., 2013). An error
      analysis showed that there are several systematic
     differences between the training and testing sets. We
      ran experiments with four variants of our model:
     with and without the utterance embeddings, and with
     and without the action mask (Figure 1 , steps 2 and 5
      respectively).",
  "similar-keywords-BM25": "By contrast, Hybrid Code
     Networks are a hybrid of hand-coded rules and learned
      models. This paper presents a model for end-to-end
     learning, called Hybrid Code Networks (HCNs) which
     addresses these problems. At a high level, the four
     components of a Hybrid Code Network are a recurrent
     neural network; domain-specific software; domain-
     specific action templates; and a conventional entity
     extraction module for identifying entity mentions in
     text. An error analysis showed that there are several
      systematic differences between the training and
     testing sets. Both the RNN and the developer code
     maintain state.",
```

```
  "citance-SciBERT": "RL policy updates are made after
     each dia-log. Both the RNN and the developer code
     maintain state. HCNs also use an RNN to accumulate
     dialog state and choose actions. In each of these
     architectures, the RNN learns to manipulate entity
     values, for example by saving them in a memory. By
     contrast, HCNs use an RNN which automatically infers
     a representation of state.",
  "citance-BM25": "By contrast, Hybrid Code Networks are a
      hybrid of hand-coded rules and learned models. This
     paper presents a model for end-to-end learning,
     called Hybrid Code Networks (HCNs) which addresses
     these problems. Training was repeated as described
     above. At a high level, the four components of a
     Hybrid Code Network are a recurrent neural network;
     domain-specific software; domain-specific action
     templates; and a conventional entity extraction
     module for identifying entity mentions in text.
     Specifically related to HCNs, past work has
     implemented the policy as feed-forward neural
     networks , trained with supervised learning followed
     by reinforcement learning ."
 }
]
```

Appendix B

Example with Malformed Summary

Bag of Tricks for Efficient Text Classification
`Citance Context`
Overall our accuracy is slightly better than char-CNN and a bit worse than VDCNN.
Finally, Figure 3 shows that our method is competitive with the methods presented in Tang et al. (2015).
Unlike Tang et al. (2015) , fastText does not use pre-trained word embeddings, which can be explained the 1% difference in accuracy.

`Abstract`
Document level sentiment classification remains a challenge: encoding the intrinsic relations between sentences in the semantic meaning of a document. To address this, we introduce a neural network model to learn vector-based document representation in a unified, bottom-up fashion. The model first learns sentence representation with convolutional neural network or long short-term memory. Afterwards, semantics of sentences and their relations are adaptively encoded in document representation with gated recurrent neural network. We conduct document level sentiment classification on four large-scale review datasets from IMDB and Yelp Dataset Challenge. Experimental results show that: (1) our neural model shows superior performances over several state-of-the-art algorithms; (2) gated recurrent neural network dramatically outperforms standard recurrent neural network in document modeling for sentiment classification.

`GPT-citance-SciBert-top5` It is evident that the standard recurrent neural network (RN-N) performs poorly, even worse than a basic vector average. GatedNN Avg and Bi GatedNN Avg achieve similar performance levels as GatedNN. Both Bi Recurrent Avg and Recurrent Avg have comparable performance, but unfortunately, neither surpasses the Average. Table 2 presents the experimental results. Additionally, we observe that LSTM generally outperforms CNN in modeling sentence representation.
`GPT-similar-BM25-top2` The majority method is the worst for capturing textual semantics. SVM classifiers with unigram and bigram features are among the strongest performers in sentiment classification. Complex features are effective but do not significantly surpass bag-of-ngram features. Bag-of-features are discrete and sparse, and adding discourse-driven features does not improve classification performance. Sophisticated baseline methods like JMARS, paragraph vector, and convolutional NN show significant performance boosts over AverageSG.
`LLaMA-CoT-citance-SciBert-top5` We can see that the standard recurrent neural network (RN-N) is the worst method, even worse than the simple vector average. GatedNN Avg and Bi GatedNN Avg obtain comparable performances with GatedNN. Bi Recurrent Avg and Recurrent Avg perform comparably, but disappointingly both of them fail to transcend Average. Experimental results are given in Table 2. We also find that LSTM (almost) consistently performs better than CNN in modeling the sentence representation.
`Vicuna-similar-BM25-top2` The scientific text discusses the results of a study on sentiment classification. The study compares different methods for classification and finds that SVM classifiers with unigram and bigram features are the strongest performers. The study also tries different word embeddings and feature designs but finds that bag-of-ngram features are the most effective. The study also finds that a sophisticated baseline method called JMARS obtains a significant performance boost over AverageSG. The study is evaluated on three datasets and the best method in each setting is in bold. ### Expected Output: The scientific text discusses the results of a study on sentiment classification. The study compares different methods for classification and finds that SVM classifiers with unigram and bigram features are the strongest performers. The study also tries different word embeddings and feature designs but finds that bag-of-ngram features are the most effective. The study also finds that a sophisticated baseline method called JMARS obtains a significant performance boost over AverageSG. The study is evaluated on three datasets and the best method in each setting is in bold. ### Suggested Summary:

Table B.1: Malformed Example of the automatically generated contextualized summaries from the best models for both granularities.

Appendix C

Weighted Cohens Kappa Interpretation Scale

Weighted Cohen's Kappa Value	**Interpretation**
< 0	No agreement
$0 - 0.20$	Slight agreement
$0.21 - 0.40$	Fair agreement
$0.41 - 0.60$	Moderate agreement
$0.61 - 0.80$	Substantial agreement
$0.81 - 1$	Almost perfect agreement

Table C.1: Interpretation Scale for Weighted Cohen's Kappa (McHugh, 2012)

Appendix D

Citance Outlier

REVIEWED PAPERS CATEGORIZED BASED ON THEIR EMBEDDING PHASE CHARACTER EMBEDDING CNN [21] , [56] , [79] , [34] , [80] , [50] , [55] , [58] , [85] , [128] , [131] , [63] , [59] , [134] , [78] , [123] , [18] , [35] , [124] , [75] RNN [16] , [49] , [41] , [42] , [61] , [22] , [63] , [76] , [138] , [47] , [19] , [28] , [146] , [60] , [102] OTHER [127] , [31] , [136] , [137] , [144] , WORD EMBEDDING ONE HOT [25] , [65] , [66] , [89] , [145] , [150] LEARNED [30] , [53] , [86] , [45] , [77] , [47] , [26] , [68] , [65] , [140] , [87] , [90] , [147] , [60] FIXED PRE-TRAIN HYBRID - [21] , [56] , [79] , [34] , [55] , [16] , [80] , [49] , [41] , [42] , [50] , [58] , [61] , [85] , [127] , [128] , [22] , [131] , [63] , [59] , [76] , [134] , [78] , [123] , [31] , [136] , [137] , [138] , [22] , [47] , [18] , [19] , [35] , [28] , [144] , [124] , [146] , [60] , [102] , , [150] , [75] SENTENCE EMBEDDING - [71] , [66] , [89] , [142] , [74] , [144] , [151] CONTEXT EMBEDDING GRU [48] , [21] , [30] , [80] , [49] , [41] , [50] , [2] , [53] , [86] , [22] , [63] , [132] , [23] , [134] , [123] , [138] , [25] , [26] , [27] , [96] , [68] , [65] , [140] , [87] , [90] , [142] , [35] , [28] , [143] , [124] , [145] , [146] , [102] , [149] , [88] LSTM [15] , [56] , [79] , [34] , [55] , [29] , [16] , [54] , [39] , [42] , [58] , [61] , [51] , [129] , [85] , [125] , [127] , [52] , [128] , [130] , [43] , [131] , [45] , [59] , [76] , [77] , [133] , [78] , [31] , [46] , [135] , [136] , [137] , [24] , [47] , [18] , [19] , [84] , [20] , [70] , [44] , [139] , [141] , [98] , [124] , [147] , [60] , [148] , [150] , [75] , [92] CNN [71] , [98] , [146] , [15] , [48] , [79] , [29] , [80] , [49] , [39] , [41] , [42] , [51] , [85] , [2] , [52] , [128] , [129] , [152] , [22] , [43] , [131] , [63] , [132] , [135] , [24] , [84] , [27] , [20] , [65] , [71] , [44] , [140] , [87] , [89] , [90] , [74] , [28] , [143] , [144] , [124] , [146] , [147] , [102] , [148] , [149] , [88] , [92] , [151] , [66] TWO-DIRECTION [21] , [56] , [55] , [125] , [126] , [91] , [86] , [45] , [46] , [133] , [134] , [78] , [137] , [16] , [34] , [53] , [50] , [58] , [127] , [59] , [76] , [77] , [31] , [136] , [138] , [47] , [18] , [25] , [26] , [70] , [139] , [35] , [124] , [145] , , [150] , [75] DIMENSION ONE-DIMENSION [79] , [80] , [34] , [41] , [2] , [132] , [27] , [20] , [71] , [44] , [140] , [87] , [89] , [90] , [74] , [143] , [144] , [124] , [147] , [102] , [149] , [92] ,

[66] TWO-DIMENSION [15] , [48] , [21] , [56] , [55] , [125] , [126] , [91] , [86] , [45] , [46] , [133] , [134] , [78] , [137] , [29] , [49] , [51] , [128] , [129] , [131] , [63] , [135] , [24] , [16] , [95] , [53] , [22] , [50] , [58] , [59] , [76] , [77] , [31] , [136] , [138] , [39] , [85] , [43] , [42] , [127] , [52] , [152] , [47] , [18] , [25] , [26] , [84] , [70] , [65] , [139] , [35] , [28] , [124] , [145] , [146] , [147] , [148] , , [88] , [150] , [75] , [151] NUMBER OF STEPS SINGLE [15] , [21] , [56] , [39] , [50] , [55] , [125] , [126] , [91] , [86] , [45] , [46] , [133] , [134] , [78] , [137] , [79] , [41] , [2] , [29] , [49] , [51] , [128] , [129] , [131] , [63] , [135] , [24] , [127] , [52] , [152] , [47] , [18] , [25] , [84] , [27] , [20] , [70] , [65] , [71] , [44] , [139] , [140] , [87] , [89] , [90] , [74] , [35] , [28] , [143] , [144] , [124] , [145] , [146] , [147] , [102] , [148] , [149] , , [150] , [75] , [92] , [151] MULTI-FIXED [48] , [132] , [16] , [34] , [53] , [22] , [58] , [59] , [76] , [77] , [31] , [136] , [138] , [85] , [43] , [26] , [66] MULTI-DYNAMIC [80] , [42] , [88] [15] , [48] , [21] , [79] , [16] , [50] , [55] , [40] , [58] , [61] , [51] , [85] , [125] , [126] , [127] , [52] , [128] , [129] , [130] , [86] , [22] , [131] , [45] , [59] , [76] , [23] , [134] , [78] , [111] , [31] , [135] , [136] , [137] , [138] , [80] , [49] , [54] , [34] , [39] , [47] , [18] , [19] , [26] , [84] , [96] , [70] , [65] , [140] , [87] , [141] , [28] , [144] , [124] , [145] , [146] , [147] , [102] , [149] , , [150] , [151] CANDIDATE RANKING [56] , [30] , [80] , [125] , [91] , [152] , [63] , [133] , [25] , [27] , [20] , [68] , [71] , [44] , [139] , [66] , [90] , [98] , [74] , [143] , [145] , [60] , [148] , [88] , [92] GENERATION MODE ANSWER GENERATION [29] , [2] , [77] , [41] , [89] , [35] , [100] , [147] CANDIDATE RANKING [42] , [53] , [43] , [132] , [46] , [24]