Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 91–100
DOI: 10.15439/2021F128
ISSN 2300-5963 ACSIS, Vol. 25

Query Specific Focused Summarization of Biomedical Journal Articles

Akshara Rai, Suyash Sangwan, Tushar Goel, Ishan Verma, Lipika Dey
TCS Research
New Delhi, India
Email: (akshara.rai, suyash.sangwan, t.goel, ishan.verma, lipika.dey)@tcs.com

***Abstract*—During COVID-19, a large repository of relevant literature, termed as "CORD-19", was released by Allen Institute of AI. The repository being very large, and growing exponentially, concerned users are struggling to retrieve only required information from the documents. In this paper, we present a framework for generating focused summaries of journal articles. The summary is generated using a novel optimization mechanism to ensure that it definitely contains all essential scientific content. The parameters for summarization are drawn from the variables that are used for reporting scientific studies. We have evaluated our results on the CORD-19 dataset. The approach however is generic.**

***Index Terms*—Extractive Summarization, Query Answering, Biomedical Text-mining, Scientific Repositories, CORD-19 dataset**

I. Introduction

With the rapid rise of scholarly articles in the biomedical domain, there has been a growing urgency to explore Natural Language Processing (NLP) techniques that can process vast volumes of content to generate intelligent insights, which can then be selectively explored by the experts. This was proved once again during the current COVID-19 pandemic. There has been a stupendous rise in related biomedical articles that have been published over the period. While it undoubtedly helped medical practitioners, virologists, immunologists, policy makers, public health planners, drug manufacturers and many others associated to healthcare services, it also highlighted the need for efficient mechanisms to enable intelligent navigation through this sea of content. The needs of end-users can be quite varied in nature. For example, in the current scenario, while medical professionals need insights about drugs and procedures, a virologist would be interested in studying the nature of the virus and hence look for literature reporting the virus's transmission, incubation, susceptibility to external factors, etc. Public policy makers, on the other hand, need information to design effective policies and guidelines to keep the spread controlled. Since, time is premium for every user, a mechanism that will enable the user to grasp the key aspects covering objectives, methodology and findings or outcomes, if any, of an article is an important ask from the NLP community.

In May 2020, as Allen Institute shared a large repository called "CORD-19"[1], which contained bio-medecial articles related to corona virus. Kaggle further announced a challenge, in which some of the key questions asked by the end-users were put up for the natural language processing community to find out efficient methods to answer them. A two-way communication ensued on the platform between the end-users and the NLP researchers, wherein the focus was to understand the requirements clearly. The discussion led to clearer elicitation of information components from different categories of users. As it turned out, while the information components were different for different category of users, all users wanted to view the relevant findings about the components in a contextual way, that would make it easy for them to interpret the significance of the results. For example, epidemiologists specifically wanted to know the "incubation period" of the virus, in order to design policies for prevention and control. However, as different values for incubation period were reported by scientists from different corners of the world, the epidemiologist wanted the result to be presented along with its context that included the sample type, sample size and most importantly the statistical outcomes of these results. The contextual presentation was clearly needed to help them decide whether to accept or reject the results. Similarly, a doctor may want to know about the drugs that were found to be effective, but along with it also the details about patient condition and treatment course, to help in decision making. It has to be further remembered that a single article may contain information that could be of interest to multiple categories of users, though all of it may not be of interest to any one category. Though the requirements were first published in Kaggle, subsequently, TREC also posted similar requirements from the CORD-19 collection. For a large number of short queries, it posted additional narratives stating stricter requirements for a retrieved article to qualify as relevant. It was observed that the narratives were similar to the user requirements mentioned in the Kaggle platform.

Motivated by the above requirements, in this paper, we present a mechanism that can create a query-specific contextually focused summary of an article for the end-user. The rationale of the proposed mechanism comes from commonly followed reporting style for bio-medical articles, especially for reporting experimental studies and case studies. The target of our work is to generate a uniformly-structured summary that contains all relevant information for a specific end-user. Thus, two end-users, based on their requirements, may see two

[1] https://allenai.org/data/cord-19

different summaries of the same document, though both the summaries will be structured in a similar fashion. Section II presents more details about the structure of an ideal summary.

This is achieved in three stages.

- We first provide a query representation mechanism that can accommodate the user requirements in terms of 5 parameters that comprise key aspects of a scientific study: *study type, sample size, sample type, measures/results, evidence of measure*. The rationale for selecting these five parameters is explained in detail in section III.
- Next, an optimization-driven method is proposed to select a minimal set of sentences that can satisfy the requirements of a query. It is done by scoring the sentences based on their information content with respect to the above-mentioned parameters, with additional constraints imposed on their proximity. The proximity constraints have been designed based on commonly followed practices for reporting outcomes in bio-medical scientific publications. These sentences form a "snippet", which can provide the key outcomes at a glance. This is explained in section IV-C.
- Finally, a contextual summary creation method is proposed. The contextual summary is created by rearranging the set of sentences selected by the optimizer and augmenting them with additional content, if necessary, to create a cohesive and comprehensive summary. This is explained in section IV-D.

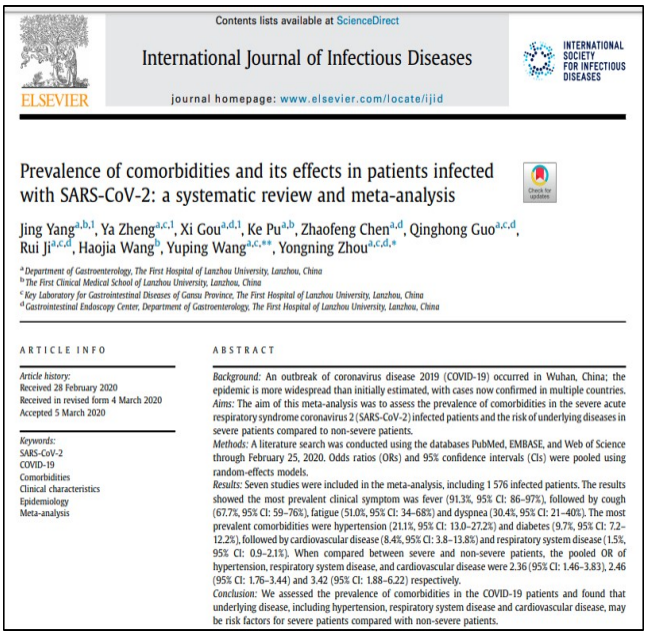
  The proposed approach ensures that the necessary information components found in the documents are always contained in the summary.

In the absence of any gold-standard data-set for evaluating the contextually focused summaries created by the proposed method, we have evaluated the summaries by comparing them with the abstracts provided along with the articles. We show that, for journals that insist on a structured summary for authors, the generated summaries are very similar to author-provided summaries. However, such journals are very few. Thus only 25% articles in the repository were found to have high-quality author-generated structured summaries. The focused summary generation method can thus be used to generate high quality summaries for a larger collection of bio-medical articles. This, by itself, is a very significant contribution to the domain of bio-medical literature analysis. The results and observations are discussed in detail in section V.

It may be noted that, the proposed mechanism is not an alternative to online document search systems which pull documents from an indexed collection in response to a query. Rather, our work is intended to augment the search results by generating a query specific summary for all articles retrieved by the search engine in response to a query. Subsequently, documents are re-ranked based on the quality of the summary. The contextual summary can be shown as a snippet to the end-user for faster comprehension.

A summary of related work in the allied area has been presented in section VI.

II. Structure of an Ideal Summary of a Bio-medical Article



Contents lists available at ScienceDirect

International Journal of Infectious Diseases

journal homepage: www.elsevier.com/locate/ijid

ELSEVIER

INTERNATIONAL SOCIETY FOR INFECTIOUS DISEASES

Prevalence of comorbidities and its effects in patients infected with SARS-CoV-2: a systematic review and meta-analysis

Jing Yang[a,b,1], Ya Zheng[a,c,1], Xi Gou[a,d,1], Ke Pu[a,b], Zhaofeng Chen[a,d], Qinghong Guo[a,c,d], Rui Ji[a,c,d], Haojia Wang[b], Yuping Wang[a,c,**], Yongning Zhou[a,c,d,*]

[a] Department of Gastroenterology, The First Hospital of Lanzhou University, Lanzhou, China
[b] The First Clinical Medical School of Lanzhou University, Lanzhou, China
[c] Key Laboratory for Gastrointestinal Diseases of Gansu Province, The First Hospital of Lanzhou University, Lanzhou, China
[d] Gastrointestinal Endoscopy Center, Department of Gastroenterology, The First Hospital of Lanzhou University, Lanzhou, China

ARTICLE INFO

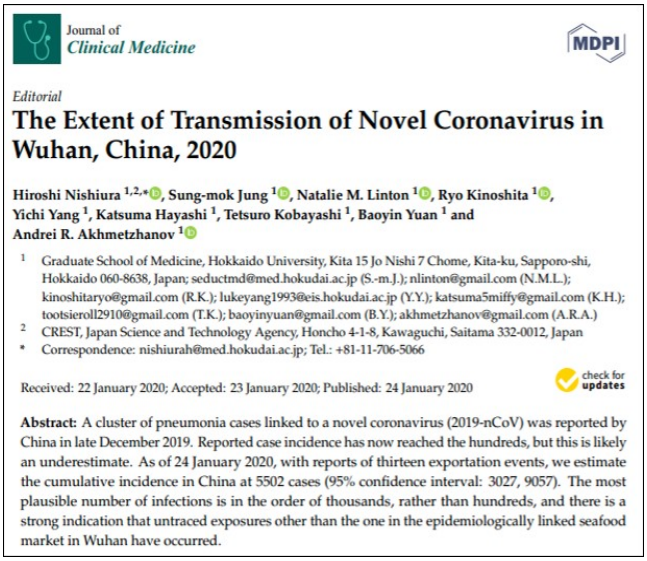
Article history:
Received 28 February 2020
Received in revised form 4 March 2020
Accepted 5 March 2020

Keywords:
SARS-CoV-2
COVID-19
Comorbidities
Clinical characteristics
Epidemiology
Meta-analysis

ABSTRACT

Background: An outbreak of coronavirus disease 2019 (COVID-19) occurred in Wuhan, China; the epidemic is more widespread than initially estimated, with cases now confirmed in multiple countries.
Aims: The aim of this meta-analysis was to assess the prevalence of comorbidities in the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) infected patients and the risk of underlying diseases in severe patients compared to non-severe patients.
Methods: A literature search was conducted using the databases PubMed, EMBASE, and Web of Science through February 25, 2020. Odds ratios (ORs) and 95% confidence intervals (CIs) were pooled using random-effects models.
Results: Seven studies were included in the meta-analysis, including 1 576 infected patients. The results showed the most prevalent clinical symptom was fever (91.3%, 95% CI: 86–97%), followed by cough (67.7%, 95% CI: 59–76%), fatigue (51.0%, 95% CI: 34–68%) and dyspnea (30.4%, 95% CI: 21–40%). The most prevalent comorbidities were hypertension (21.1%, 95% CI: 13.0–27.2%) and diabetes (9.7%, 95% CI: 7.2–12.2%), followed by cardiovascular disease (8.4%, 95% CI: 3.8–13.8%) and respiratory system disease (1.5%, 95% CI: 0.9–2.1%). When compared between severe and non-severe patients, the pooled OR of hypertension, respiratory system disease, and cardiovascular disease were 2.36 (95% CI: 1.46–3.83), 2.46 (95% CI: 1.76–3.44) and 3.42 (95% CI: 1.88–6.22) respectively.
Conclusion: We assessed the prevalence of comorbidities in the COVID-19 patients and found that underlying disease, including hypertension, respiratory system disease and cardiovascular disease, may be risk factors for severe patients compared with non-severe patients.

Fig. 1: Structured abstract



Journal of Clinical Medicine

MDPI

Editorial

The Extent of Transmission of Novel Coronavirus in Wuhan, China, 2020

Hiroshi Nishiura [1,2,*], Sung-mok Jung [1], Natalie M. Linton [1], Ryo Kinoshita [1], Yichi Yang [1], Katsuma Hayashi [1], Tetsuro Kobayashi [1], Baoyin Yuan [1] and Andrei R. Akhmetzhanov [1]

[1] Graduate School of Medicine, Hokkaido University, Kita 15 Jo Nishi 7 Chome, Kita-ku, Sapporo-shi, Hokkaido 060-8638, Japan; seductmd@med.hokudai.ac.jp (S.-m.J.); nlinton@gmail.com (N.M.L.); kinoshitaryo@gmail.com (R.K.); lukeyang1993@eis.hokudai.ac.jp (Y.Y.); katsuma5miffy@gmail.com (K.H.); tootsieroll2910@gmail.com (T.K.); baoyinyuan@gmail.com (B.Y.); akhmetzhanov@gmail.com (A.R.A.)
[2] CREST, Japan Science and Technology Agency, Honcho 4-1-8, Kawaguchi, Saitama 332-0012, Japan
* Correspondence: nishiurah@med.hokudai.ac.jp; Tel.: +81-11-706-5066

Received: 22 January 2020; Accepted: 23 January 2020; Published: 24 January 2020

check for updates

Abstract: A cluster of pneumonia cases linked to a novel coronavirus (2019-nCoV) was reported by China in late December 2019. Reported case incidence has now reached the hundreds, but this is likely an underestimate. As of 24 January 2020, with reports of thirteen exportation events, we estimate the cumulative incidence in China at 5502 cases (95% confidence interval: 3027, 9057). The most plausible number of infections is in the order of thousands, rather than hundreds, and there is a strong indication that untraced exposures other than the one in the epidemiologically linked seafood market in Wuhan have occurred.

Fig. 2: Unstructured abstract

A well-structured summary is expected to contain all required information in a compact, cohesive and comprehensible fashion. Though scientific documents usually contain abstracts that present a short and concise summary of the document, our analysis of the CORD-19 collection revealed that abstracts vary widely in size and nature, depending on the journal in which it is published. We observed that bio-medical documents contain two types of abstracts, i.e. 'Structured abstract' and 'Unstructured abstract'. Structured abstracts usually present a

well-defined and detailed summary of the document. Figure 1 shows an example of a structured abstract [1], where *Background, Method, Results and Conclusion* of the experiment are separately presented in the abstract itself. Unstructured abstracts, as shown in Figure 2 [2], on the other hand are generally short and may not convey all the important elements included in the introduction, method, or findings sections. Both these abstracts were created by the respective authors, who selected which information goes to the abstract and which does not. In the absence of a strict requirement, the author-created abstract may or may not contain the information that is required by a user, even though it may be contained in the article.

The proposed work intends to cover this gap by providing a mechanism to create focused well-structured summaries on the fly, which will contain the user-required information, if it is there in the document. These summaries should be similar in form to the structured abstracts shown in Figure 1. In order to do that, we exploit the inherent structure that is observed in the published articles. Bio-medical articles usually follow a specific format for reporting their findings. The findings are usually reported along with additional details about (a). the type of the study or the way the experiment or study was conducted (b). details about the subject of the experiment i.e. about the sample types, categorization of the samples, sample size etc. (c). results of experiments or observations (d). evidence of measure for different sample categories (e). the significance of the results. There is also a discipline that is maintained while reporting these items. For example, significance of a result is explained along with evidence of measure.

In the next section, we first present a few sample queries published on the Kaggle site along with the requirements of each. Subsequently, we discuss how these requirements can be mapped to the scientific parameters and converted to a slot-value format, which is used later to construct optimization constraints. The optimizer then uses these constraints to select an optimal set of sentences that can satisfy the user requirements.

III. Query representation mechanism for summary construction

Table I shows four types of questions, posted under different task categories in the CORD-19 [2] challenge by various groups of users. Each question is accompanied by a narrative that specifies what kind of information is required from the documents, to answer the queries. These four questions represent four broad and exhaustive categories, which cover most of the user queries posed to the collection. We now present a mapping of these queries to the parameter requirements mentioned earlier. The mapping is done to five different slots that can be associated to specific types of values.

1) *Study Type:* describes a broad category for the type of work reported in the document. It could be a systematic review, a case study or case series, a simulation study or an experimentation. This covers almost all kind of documents, but more may be added.
2) *Sample Size:* is used to define the size of the study population, samples studied or papers reviewed to compute the result. For example, 50 Patients, 120 case reports, etc.
3) *Sample Type:* describes the sub sample of the population addressed or the type of samples that were considered for the study. For example, population addressed can be pregnant women, children, elderly, smokers, etc.
4) *Measures/Results:* These are the quantitative outcomes or findings presented in a document after analysis of the data. They can be statistical findings like odds ratio, hazard risk, etc. on potential risks or other outcomes like drug effectiveness, prevalence, etc.
5) *Evidence of Measure:* These are additional qualifiers or filters that are applied on the measures/results to quantify the level of evidence. Evidences can be expressed in terms of sets of sub-samples generated from the population. For example, the risk posed by COVID-19 to smokers can vary depending on their age and other co-morbidities present. The impact of a policy or guideline depends on the country it is implemented at. Thus, these elements can be used to present the evidence of measure of various queries.

Table II presents a few sample user queries from Kaggle site, along with their mapping to the question type presented in Table I, further slotted according to the type of information required. The slot-value requirements for each question type is derived from the narratives. This is further validated using the target requirements mentioned for these queries at the Kaggle site.

Slot items are associated with factor-specific constraints that are designed to ensure that only meaningful information components are picked up. For example, odds ratio is usually specified in a paper as "OR <INTEGER>, 95% CI <RANGE>", incubation period is presented as "number of days", country names can only be from a set of known entities, drug names can be recognized using Biological entity taggers. Each slot is also associated to an encapsulated information extraction procedure which hunts for feasible values for that slot. Table II also gives some examples of accepted study design types for the bio-medical domain. A list of such constraints has been curated from available literature and data on the challenge sites. This list can be extended.

In order to ensure the coverage of queries using these categories of questions and slot types, we have additionally considered the queries presented by the TREC challenge makers[3] to be addressed from the CORD-19 collection. We were able to map approximately 67% queries to these 4 broad categories mentioned in Table I and further identify the slot requirements on the basis of the narrative. For example,

[2]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks

[3]we have used the topic set from Round 1 data - https://ir.nist.gov/covidSubmit/data.html

Type	Category	Kaggle Questions	Detailed Requirement
1	Risk Factors	What do we know about COVID-19 risk factors?	Data on potential risks factors: Smoking, pre-existing pulmonary disease, Co-infections and other co-morbidities. Severity of disease, including risk of fatality among symptomatic hospitalized patients, and high-risk patient groups. Susceptibility of this specific population. Mitigation measures that could be effective for control
2	Epidemiological Requirements/Clinical characteristics	What is known about transmission, incubation, and environmental stability?	What do we know about natural history, transmission, and diagnostics for the virus? What have we learned about infection prevention and control? Range of incubation periods for the disease in humans (and how this varies across age and health status) and how long individuals are contagious, even after recovery. Prevalence of asymptomatic shedding and transmission. Persistence and stability on a multitude of substrates and sources (e.g., nasal discharge, sputum, urine, fecal matter, blood). Persistence of virus on surfaces of different materials (e,g., copper, stainless steel, plastic).
3	Treatment/Diagnostics Efficacy	What do we know about vaccines and therapeutics?	Effectiveness of drugs being developed and tried to treat COVID-19 patients. Clinical and bench trials to investigate less common viral inhibitors against COVID-19. Capabilities to discover a therapeutic (not vaccine) for the disease, and clinical effectiveness studies to discover therapeutics, to include antiviral agents. Use of diagnostics such as host response markers (e.g., cytokines) to detect early disease or predict severe disease progression, which would be important to understanding best clinical practice and efficacy of therapeutic interventions.
4	Non Pharmaceutical Intervention/ Relevant External Factors	What do we know about non-pharmaceutical interventions and the Relevant factors related to COVID-19	Rapid design and execution of experiments to examine and compare NPIs currently being implemented. Rapid assessment of the likely efficacy of school closures, travel bans, bans on mass gatherings of various sizes, and other social distancing approaches. Methods to control the spread in communities, barriers to compliance and how these vary among different populations. Models of potential interventions to predict costs and benefits that take account of such factors as race, income, disability, etc. Seasonality of transmission, How does temperature and humidity affect the transmission of 2019-nCoV? Significant changes in transmissibility in changing seasons? Effectiveness of personal protective equipment (PPE)

TABLE I: User given questions and their detailed requirements.

	Inferred Slot Requiremnts					
Sample User Query	Query Type	Study Type	Sample Size	Sample Type	Measures/Results	Evidence of Measure
Risk to pregnant women	1	Systematic review, case series	#patients	Pregnant women	Odds ratio, hazard ratio, severity	-
Incubation period of Sars-Cov-2	2	Simulation, meta-analysis	#patients	-	No. of Days/weeks	age, gender
Effectiveness of Remdesivir in treating COVID-19	3	RCT, systematic review, meta-analysis	#patients	patients treated with Remdesivir	Odds ratio, hazard ratio, severity	Therapeutic method(s) utilized/assessed
Effect of social distancing in reducing virus spread	4	Simulation, Cross-sectional study, systematic review	-	Population(general, healthcare, minority)	Percentage Decrease/Increase, mortality rate, days	Intervention: Social Distancing, Geographical location, model used

TABLE II: Slotted user requirements for sample Kaggle queries

the TREC query- *'Are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk for COVID-19?'* can be mapped to question Type 1 with the following constraint - <Sample Type, Patients taking ACE inhibitors>. Similarly, another query *'How long can the coronavirus live outside the body'* can be assigned to Type 2 with <Measure/Results, Persistence(in days, hrs, half life)> and <Evidence of Measure, (Sample Observed, Detection Method)> slot requirements. Table III shows a few examples of queries from the TREC dataset.

The remaining 33% queries required theoretical evidence based excerpts, such as - *'What are best practices in hospitals and at home in maintaining quarantine?', 'How has lack of testing availability led to under reporting of true incidence of Covid-19?'*.

A. Ensuring consistency of information

After identifying the required slots, they are further bucketed together to ensure meaningful information extraction. The buckets represent groups of slot items that are inter-dependent on each other with respect to the given query. The inter-dependence of these items is either expressed as a linguistic constraint or a proximity constraint. These constraints are also parsed from the narrative. For example, for a query "what is the range of incubation period for different age groups?" the slot value pairs are filled up as <Measure/Results, incubation period>, <Evidence of measure, age group> and <Sample size, #patients>. This in turn implies that the statement "The average incubation period was 4 days," found in a document wouldn't be complete. It needs additional information for the result to be accepted. A sentence found in close proximity to the above one was "We considered 157 confirmed cases, aged 44-60 years, 74 female (47.1%) and 38 imported cases (24.2%)." A complete snippet would have to contain both the sentences. By adding Measures/Results along with the Evidence of Measure in a single bucket, we can generate a more comprehensive and coherent snippet for the user query. Additionally, information about whether it was a simulation experiment or a systematic review, i.e. the study type of the document is also presented in the snippet. This is independent of the final result being reported and is therefore added in a separate bucket. Thus, there can be two buckets in the ar-

	Inferred Slot Requiremnts					
Sample TREC Question & Narrative	Query Type	Study Type	Sample Size	Sample Type	Measures/Results	Evidence of Measure
Are patients taking Angiotensin-converting enzyme inhibitors (ACE) at increased risk: interactions between coronavirus and angiotensin converting enzyme 2 (ACE2) receptors, risk for patients and recommendations for these patients.	1	Systematic review, case series, Retrospective	#patients	patients taking ACE	Odds ratio, hazard ratio, severity	-
How long does coronavirus remain stable on surfaces: SARS-CoV-2's virus's survival in different environments (surfaces, liquids, etc.) outside the human body while still being viable for transmission to another human	2	Experimental study, Systematic review,meta-analysis	#samples	-	No. of Days/weeks, half life	Surfaces, # studies, Method used
What types of rapid testing for Covid-19 have been developed? : ways to diagnose Covid-19 more rapidly	3	clinical trial, retrospective, systematic review, meta-analysis	#patients	infected patients	Efficiency, speed of Assay	Detection Method: Rapid
How does the coronavirus respond to changes in the weather: virus viability in different weather/climate conditions, transmission of the virus in different climate conditions	4	Simulation, Cross-sectional study, systematic review, retrospective	-	-	Incidence, transmission, mortality rates	External Factor: weather, Geographical location, model used

TABLE III: Sample TREC questions and narratives with slotted user requirements

rangement to capture all the slots. Bucket 1 contains (Evidence of Measure, Measures/Result) and Bucket 2 includes (Sample type, Sample Size, Study Type).

The buckets can be interpreted as context provider for the information components to ensure that randomly occurring strings or values of a certain type are not accepted just because of a keyword match.

IV. Query Focused Summary Generation

The task of of generating query specific focused summaries is carried out in phased manner. Initially, the slot-value pairs are searched within the document collection. This is done by locating the entities within the document. Each document is subjected to a pre-processing phase for the purpose. After that, a set of minimal number of sentences is selected that satisfies the slot requirements, with additional constraints imposed on the proximity of slots within a single bucket, using integer linear programming. A subsequent phase of enhanced document summarization is carried out to present the information in a coherent and comprehensible form.

A. Document Pre-processing

Like all document processing tasks, search for information is preceded by a one-time activity that comprises of document pre-processing and information extraction. Each document in the set is passed through a pre-processing pipeline for cleaning and tokenizing it into sentences using SciSpaCy [3]. Each sentence is then indexed according to its unique document-id and the section label where it belongs in the document. Each document is then subjected to the following processes-

- **Biomedical Entity Extraction**: Given the biomedical documents, this module extracts biomedical entities like Participant Age, Participant Sex, Participant Sample size, Participant Condition, Surgical Intervention, Physical Intervention, Educational Intervention, Psychological Intervention, Control Intervention, Outcome Physical, Pain Mentions, Mortality Mentions, Mental States and Adverse effects. These entities are extracted using a BERT-based sequence labelling approach described in [4]. Additionally, biomedical entities like DNA, Cell Type, Protein, Chemical, Organ names, Drug, etc. are also extracted using SciSpaCy.
- **Named Entity Extraction**: Named entities like name of the locations, person, organizations, expressions of quantities ('0.2 ng/mL'), time ('less than 24 hours'), age ('49 years old', 'one week old') are extracted from each document using SpaCy [5].
- **Sentence embedding generation**: Sentence embeddings are also generated using Facebook's Infersent pre-trained encoder [6] to create a 300-dimensional vector for a sentence. It uses Bidirectional LSTM with max pooling to capture the context and generic information available for a variety of tasks. These embeddings capture the semantics of a sentence better by embedding the context in the encoding.

B. Mechanism for sentence scoring

In this section, we present how the specific information components required for a query are located within the documents and scored to generate a snippet. First, the sentences are checked for the presence of any of the required slot values. Slot specific search methods are deployed for this purpose. The extraction methods commonly used for the different slots are as follows: -

1) *Measures/Results*- As observed from the summary tables provided by the CORD-19 challenge makers, values fitting this slot (like OR, p-value, HR, etc) follow a

set pattern, which can be expressed using a regular expression such as "<MeasureName>= <INTEGER>(, 95% CI <RANGE>)?". At first, we used a regular expression matching algorithm to extract instances of this type. But the pattern matching approach resulted in noisy extractions and also missed certain instances that varied slightly from this pattern. Therefore, we moved on to use a BiLSTM-CRF sequence tagger [7] to identify the measures/results in sentences, which showed an accuracy of 97%. Here, we have used the results from above pattern-matching approach along with certain hand tagged instances (that were not detected earlier) to create the annotated training data for a sequence tagger. We have excluded the noisy extractions of pattern-matching approach from the training data. Since the task is to identify a set of literals/token following a pattern, we did not use any sequence tagging algorithm requiring semantic context.

2) *Study Type*- These are pre-specified strings and keywords found in text. A comprehensive design dictionary curated by a team of epidemiologists has been provided to help the CORD-19 research community for effective retrieval. [4]
3) *Sample Size*- This is extracted by tagging *'Participant Sample Size'* instances in text using the biomedical entity extractor described in the previous section.
4) *Sample Type*- Values are extracted using the biomedical entity extraction module. For any given query, findings like patient condition, patients undergoing any surgical intervention, patients having any drug administered, etc. can be selected for this slot depending on the requirement. For example, for the query 'risk to cancer patients due to COVID-19' - <patient condition, 'Cancer'> is added to the slot. For 'effectiveness of hydroxychloroquine in treatment of COVID-19 patients' the slot-value pair <Drug, 'Hydroxychloroquine'> is added.
5) *Evidence of Measure*- Values are extracted using the biomedical and Named entity extraction modules explained in the previous section. Extractions like Patient Age, Gender, country, etc. are included in this slot.

Any sentence that contains at least one value is retained for scoring, while the remaining ones are assigned a score of 0. The final score assigned to a sentence depends on three factors, which are explained below-

Confidence score from sentence type - The section headers of the document are also taken into account while scoring sentences. Thus, sentences from "review" section score less than those coming from other sections of the document, since the latter are considered to be fundamental contributions from the document under consideration. Since, section headers are not always unambiguous, special checks are put into place to check for reference and citation patterns as well as linguistic constructs to identify such sentences. For computing the confidence value, sentences from "review" sections are penalized by a value of (ρ), and the findings fundamental to the document are rewarded with (ρ), such that 0<ρ<1.

Intra-bucket score - Sentences containing values for certain slots also gain for being in proximity of other sentences containing values in the same bucket. As a corollary, between two sentences that contain values for the same slot, the one that contains additional values for other slots belonging to the same bucket will score higher. This is referred to as intra-bucket score of a sentence.

Inter-bucket score – Sentences also gain some reward from being in proximity to other sentences that contain values for slots from other buckets. The inter bucket proximity ensures that the overall context of all the findings remains consistent.

We now present the scoring equations.

Proximity between two sentences S_i and S_j, is computed as an inverse function of the distance between the sentences in the document and also takes into account their corresponding section headers.

$$Proximity(S_i, S_j) = \frac{(1+section\_reward(i,j))}{(1+distance(S_i,S_j))} \quad (1)$$

where, distance (S_i, S_j) = abs (position (S_i) - position(S_j)), position(S_i) indicates original sentence number of S_i, and Section_reward (i,j) = 1, if the section header of sentences is same; otherwise 0.

Let V = {v_1, v_2, v_3, …, v_m} be the set of values required by the query. Then the scores for a sentence S_i having a value v_k is expressed as follows:

$$Intra\_Bucket\_Score = \sum_k (Confidence(v_k) + \sum_p (max(Proximity(S_i, S_j)))) \quad (2)$$

$\forall$ v_k, v_p ϵ V, s.t. bucket(v_p) = bucket(v_k),
$\forall$ j s.t. S_j is the closest sentence that contains a value for a slot v_p that belongs to the same bucket, including itself.

$$Inter\_Bucket\_Score = \sum_k (Confidence(v_k) + \sum_p (max(Proximity(S_i, S_j)))) \quad (3)$$

$\forall$ v_k, v_p ϵ V, s.t. bucket(v_p) $\neq$ bucket(v_k) $\forall$ j s.t. S_j is the closest sentence that contains a value for a slot v_p that belongs to a different bucket, including itself.

Score (S_i) is now computed as-

$$Score(S_i) = \alpha(Intra\_Bucket\_Score(S_i)) + (1-\alpha)(Inter\_Bucket\_Score(S_i)), \quad (4)$$

We take α >0.5 to give more weightage to the Intra_Bucket scores over the Inter_Bucket scores. The sentence score is then normalized s.t. Score (S_i) ϵ [0,1].

[4]https://docs.google.com/spreadsheets/d/1t2e3CHGxHJBiFgHeW0dfwtvCG4x0CDCzcTFX7yz9Z2E/edit#gid=1217643351

C. Optimal snippet generation

Our goal is now to use the above scores to identify the minimal set of sentences that can form a snippet.

Let us suppose that query Q has 'm' slot values divided into different buckets. Let S = {S_1, S_2,..., S_n} be the set of sentences which have a non-zero scores after scoring. The following optimization algorithm finds the minimal set of sentences that contain all the 'm' values, if present.

Let VS(i, j) = 1, if value v_j is found in S_i; otherwise 0.

Let x(i) = 1, if S_i is selected in optimal snippet and 0 otherwise

Then the objective function for the optimization problem is expressed as-

Objective Function:

$$Maximize \sum_i (x(i) * (Score(S_i) - 1)) \quad (5)$$

Subject to constraints:

$$\sum_i (VS(i,j) * x(i)) >= 1 \;\; \forall v_j \; found in \; D \quad (6)$$

$$\sum_i x(i) <= |V| \quad (7)$$

$$\sum_i x(i) >= 1 \quad (8)$$

The value (–1) is added to ensure that minimum number of sentences are finally selected. The constraint in equation 6 ensures that at least 1 sentence is picked to cover each value, provided that value is reported by the document D. Finally, equations 7 and 8 enforce that at least 1 sentence is selected from the document and maximum number of sentences selected are no more than the type of values required to address the user given query. This is solved using Integer Linear programming.

Figure 3 shows the snippet generated using the above optimization approach for two documents [8, 9], along with the slot values for the queries 'Risk to Diabetes Patient' and 'Incubation period with respect to age'. It can be seen from these examples that the individual sentences by themselves are not enough. Reporting *'Fatality rate was 11.1%'* doesn't convey the confidence of the finding. By additionally reporting <*Patient Condition, 'Diabetes'*>, <*Sample Size, '258 Patients'*> *and* <*Study Type', 'Retrospective'*>, a much better picture can be presented. The second example also highlights how the proximity constraint helps provide maximum information in minimum sentences, making it much more comprehensible.

D. Contextual focused summary generation

In this section, we present an enhanced summarization approach which generates a fixed length extractive summaries for documents, by checking for sentence representativeness along with the scores from the previous section. For each candidate sentence to be included in the summary, it's 300 – dimensional vector embedding is created using Infersent.

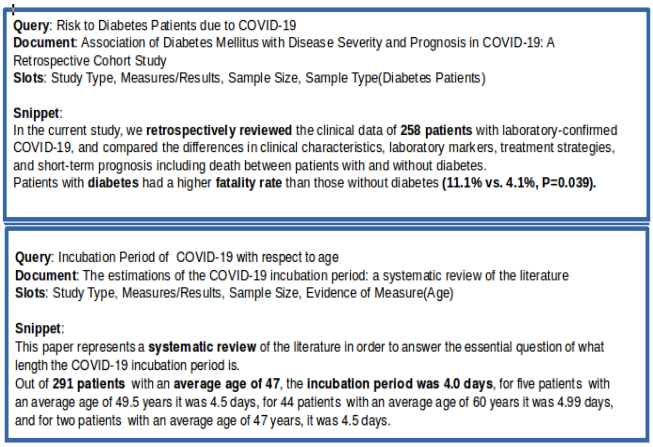


Fig. 3: Snippets generated for queries along with slot values

Sentence score (Sc) for the i^{th} sentence in the j^{th} document is generated as follows -

$$Sc(S_i^j) = Sc_{\text{Rank}}(S_i^j) + Sc_{\text{Title}}(S_i^j) + Sc_{\text{Position}}(S_i^j) + Sc_{\text{Domain}}(S_i^j), \quad (9)$$

where Sc_{Rank} (S_i^j) is the representativeness score assigned using the TextRank algorithm, by checking the sentence's similarity with all other sentences, using the corresponding Infersent vectors. Sc_{Title} (S_i^j) is computed using the cosine similarity between the title and sentence vectors. Position score proves to be very effective in document summarization as it is a good indicator of significant sentences and is computed as

$$Sc_{\text{Position}}(S_i^j) = \frac{Len_j}{Pos_i * (Len_j - Pos_i + 1)}, \quad (10)$$

where, Len_j is the length of j^{th} document, and Pos_i is the position of i^{th} sentence in the document.

Sc_{Domain} (S_i^j) denotes the score computed in the earlier section based on the slot requirements. All these scores are normalized and added to give us the final sentence score.

In order to remove redundancy, we use an algorithm similar to the MMR algorithm [10], that focuses on ensuring diversity in the sentences being selected. The sentences are sorted based on the decreasing value of their scores Sc (S_i^j) and the highest scored sentence is selected to be included in the final summary first. The next sentences are selected based on the following conditions:

Sentences are added to the final summary, iff the cosine similarity of the sentence with the selected set of sentences is below a threshold β.
Sentences having similarity with a selected sentence greater than the threshold β are discarded if they belong to the same section in the document.

This process is repeated for all the remaining sentences, till selected sentence count reaches a maximum count τ.

To ensure that the summaries are connected and coherent, the selected sentences are re-ordered according to their position in the document. Preserving document order guarantees that the summary has sentences from the aim and introduction presented first, followed by the methodology and finally, the results and conclusions.

V. Experiments and Evaluations

A. Dataset description

The Covid-19 Open Research Dataset (CORD-19) is a collection of scientific papers on Covid-19, SARS-CoV-2, and related historical coronaviruses. The dataset contains a primary metadata file containing unique paper id, author, journal, publication date, abstract etc. and link to full-text file name. Full texts are available for some files in json format.

B. Snippet evaluation and observations

We have conducted the evaluation of the snippet generation system on recently-published articles from the CORD-19 dataset. Due to lack of gold standard data, the evaluation was done manually for 10 queries across 4 categories (Table I), on 500 documents. We consider Study type, Sample Size, Evidence of Measure, Sample Type and Measures/Results as the required slots and compare the findings with the values reported in the abstracts, also measuring the overall correctness with respect to the document as well. The manual inspection of generated snippets with respect to the documents showed that study type and sample size were retrieved correctly 70.2% and 67.4% times respectively. Out of these, it was observed that 82.24% and 66.52% of the times these values matched with study type and sample size reported in the abstract. For measures/results (i.e. the quantitative findings), we evaluate them as correct if the extraction is reported in association with the user query/keyword. We observed that of the 73.6% correctly extracted values only 26.3% of the snippet values matched with the findings in the abstract. In 47.5% cases we observed that the abstracts either did not report any statistical findings or reported findings were not relevant to the query. This could be because the main theme of the document was different from that of the query. This further emphasizes the need for generating snippets and summaries from documents that answer the user queries.

C. Evaluating summaries- results and observations

We use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [11] scores for evaluating the summaries. It determines the quality of a summary automatically, by comparing it to human (ideal) generated summaries (we use the abstracts as model summaries here). ROUGE-N (unigram and bigram match) and ROUGE-L (Longest Common Subsequence match) scores were chosen for our experiments.

The generated summaries were grouped based on the type of abstract (structured and unstructured) in the document. We observed that only 145 documents (25.3%), out of 573 scientific documents summarized, had structured abstracts, remaining documents either had no abstract or had an unstructured one.

We have generated two different types of summaries using TextRank algorithm, as shown in the Table IV. In the baseline approach, we have generated generic summaries, using $Sc_{Rank}(S_i^j)$, Sc_{Title} (S_i^j), and $Sc_{Position}$ (S_i^j) scores. But in our final approach (i.e. Contextual focused summary), we have incorporated user requirements by using Sc_{Domain} (S_i^j) for scoring sentences.

In order to determine the performance, results are also compared with some existing text summarization algorithms, like LSA [12] and TextRank [13]. It can be seen from Table IV that our system performs better than these summarization algorithms. There is a 6.9% increase in ROUGE-L scores after including Sc_{Domain} (S_i^j) score in case of structured abstracts. High ROUGE scores with structured abstracts indicates that the summaries generated by our method have been able to cover the important information and findings well. Unstructured abstracts, on the other hand, seldom include results or description of the methodology. By including slots like Study Design, Sample Size, Statistical Measure/Results, the summaries generated by our approach become more informative and can present facts and details that are mostly not covered in the abstracts.

Figure 4 shows the relation between number of words in abstract and the ROUGE scores for documents with unstructured abstracts. Since the longer abstracts are supposed to be more detailed and informative, it can be seen that with the increase in word count, the ROUGE scores also increase. The evaluation with low word count abstract provides a reverse indicator for measuring the quality of the summaries, as a lower overlap means that a lot of additional information has been captured in the summary as well, that was missing in the abstract.

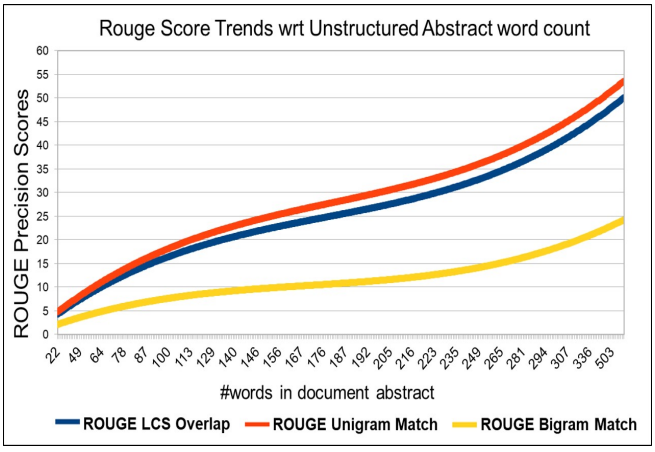


Fig. 4: ROUGE scores trends with respect to the Abstract word count

VI. Related Work

Text summarization has attracted the attention of NLP researchers for a long time. Latent Semantic Analysis (LSA) based approach was introduced in [12], which uses a singular value decomposition on word-sentence matrix. This

Approach	Documents with Structured Abstracts			Documents with Unstructured Abstracts		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.41009	0.16537	0.37369	0.29361	0.11551	0.26283
LSA	0.40491	0.11812	0.36868	0.34119	0.09503	0.30515
Baseline	0.45208	0.17312	0.41852	0.39170	0.16082	0.35838
Baseline+Domain Scores	0.47615	0.19624	0.44744	0.38696	0.15809	0.35438

TABLE IV: F-measure for summaries generated (length = 10 sentences)

way sentences that discuss important topics are chosen as candidates for summaries. One of the most successful text summarization systems called TextRank [13] was introduced in 2004. TextRank uses a graph-based algorithm similar to PageRank [14], in which similarity between two sentences is computed in terms of their content overlap. Later, [15] enhanced TextRank and proposed the use of longest common substrings based cosine distance between pairs of sentences. BM25 [16] can also be used as a ranking function to retrieve the candidate sentences for the summary. Single-document summarization approach was proposed in [17], that maximizes concept coverage using Integer Linear Programming(ILP). They also presented a weighing method for combining position to emphasize important concepts.

The information available for clinicians and clinical researchers is growing exponentially, both in the biomedical literature and patients' health records. We need strategies to cope with this information overload as biomedical literature provides clinicians and clinical researchers with a valuable source of knowledge to assess the latest advances, develop and validate new hypotheses, conduct experiments, and interpret their results [18, 19].

Several approaches have been proposed for summarization in biomedical domain. The applications mainly include summarizing treatments [20], summarizing drug information [21], summarizing clinical reports [22], and electronic health records [23]. One such work is presented in [24], a graph-based summarizer that uses the Unified Medical Language System (UMLS) to identify concepts and the semantic relations between them to construct a semantic graph that represents the document. A degree-based clustering algorithm was then used to identify different themes or topics within the text. Authors in [25] proposed a clustering and itemset mining based Biomedical Summarizer (CIBS) that also utilize UMLs to map text to concepts and then passes it to an itemset mining algorithm, for topic extraction. Sentences are clustered and related sentences from within these clusters are selected to produce a summary.

Text summarization approaches focusing on answering user queries are particularly of interest as it can aid medical practitioners identify salient and relevant information. The work in [26] presented one such approach that utilizes labeled data that is publicly available, pre-trained medical domain word embeddings along with a set of simple features for generating query focused extractive summaries.

Query-based text summarization based on common-sense knowledge and word sense disambiguation was proposed in [27]. Their technique finds semantic relatedness score between query and input text document for extracting relevant sentences. It finds correct sense of each word of a sentence with respect to the context of the sentence and hence provides query-relevant summaries.

VII. Conclusion

In this paper, we present summarization mechanism that can create a query-specific contextually focused summary of an article for the end-user. Initially, a query representation mechanism is defined that can accommodate the user requirements in terms of a fixed number of parameters that comprise key aspects of a scientific study. Further, an optimization-driven mechanism is used for retrieving minimal number of sentences relevant to an elaborate scientific query. These sentences form a snippet which provides the key outcomes at a glance. Finally, a contextual summary is created by rearranging the set of sentences selected by the optimizer and augmenting them with additional content. The target of the current work is to generate a uniformly-structured summary that contains all relevant information for a specific end-user. Thus the summaries are customized to the needs of the user. The results have been evaluated using ROUGE scores. The summaries generated by the proposed method have high ROUGE scores with the author-written summary, whenever one is present. For the remaining documents, the generated summary is a useful addition. From an application point of view, we believe that our snippet generation and summarization approach can be easily applied to other data sets by updating the slot requirements.

In future, we would like to explore more on the document structures, sentence type classification and abstractive summarization approaches for reducing the information overload even further. We also intend to extend the methods to work for any scientific document collection, beyond bio-medical literature. We are also evaluating it for a larger set of queries with enough variation in their structures and design automated evaluation mechanisms, since getting manual feedback is difficult.

References

[1] J. Yang, Y. Zheng, X. Gou, K. Pu, Z. Chen, Q. Guo, R. Ji, H. Wang, Y. Wang, and Y. Zhou, "Prevalence of comorbidities and its effects in patients infected with sars-cov-2: a systematic review and meta-analysis," *International Journal of Infectious Diseases*, vol. 94, pp. 91–95, 2020.

[2] H. Nishiura, S.-m. Jung, N. M. Linton, R. Kinoshita, Y. Yang, K. Hayashi, T. Kobayashi, B. Yuan, and A. R.

Akhmetzhanov, "The extent of transmission of novel coronavirus in wuhan, china, 2020," 2020.

[3] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.

[4] T. Dasgupta, I. Mondal, A. Naskar, and L. Dey, "Extracting semantic aspects for structured representation of clinical trial eligibility criteria," in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, Nov. 2020. doi: 10.18653/v1/2020.clinicalnlp-1.27 pp. 243–248. [Online]. Available: https://www.aclweb.org/anthology/2020.clinicalnlp-1.27

[5] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[7] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[8] Y. Zhang, Y. Cui, M. Shen, J. Zhang, B. Liu, M. Dai, L. Chen, D. Han, Y. Fan, Y. Zeng *et al.*, "Association of diabetes mellitus with disease severity and prognosis in covid-19: a retrospective cohort study," *Diabetes research and clinical practice*, vol. 165, p. 108227, 2020.

[9] N. Zaki and E. A. Mohamed, "The estimations of the covid-19 incubation period: a systematic review of the literature," *medRxiv*, 2020.

[10] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 335–336.

[11] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[12] J. Steinberger and K. Jezek, "Using latent semantic analysis in text summarization and summary evaluation," *Proc. ISIM*, vol. 4, pp. 93–100, 2004.

[13] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[15] F. Barrios, F. López, L. Argerich, and R. Wachenchauzer, "Variations of the similarity function of textrank for automated summarization," *arXiv preprint arXiv:1602.03606*, 2016.

[16] S. Robertson and H. Zaragoza, *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[17] H. Oliveira, R. Lima, R. D. Lins, F. Freitas, M. Riss, and S. J. Simske, "A concept-based integer linear programming approach for single-document summarization," in *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2016, pp. 403–408.

[18] R. Smith, "Strategies for coping with information overload," 2010.

[19] F. Davidoff and J. Miglus, "Delivering clinical evidence where it's needed: building an information system worthy of the profession," *Jama*, vol. 305, no. 18, pp. 1906–1907, 2011.

[20] H. Zhang, M. Fiszman, D. Shin, C. M. Miller, G. Rosemblat, and T. C. Rindflesch, "Degree centrality for semantic abstraction summarization of therapeutic studies," *Journal of biomedical informatics*, vol. 44, no. 5, pp. 830–838, 2011.

[21] M. Fiszman, T. C. Rindflesch, and H. Kilicoglu, "Summarizing drug information in medline citations," in *AMIA Annual Symposium Proceedings*, vol. 2006. American Medical Informatics Association, 2006, p. 254.

[22] H. Moen, L.-M. Peltonen, J. Heimonen, A. Airola, T. Pahikkala, T. Salakoski, and S. Salanterä, "Comparison of automatic summarisation methods for clinical free text notes," *Artificial intelligence in medicine*, vol. 67, pp. 25–37, 2016.

[23] R. Pivovarov and N. Elhadad, "Automated methods for the summarization of electronic health records," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 938–947, 2015.

[24] L. Plaza, A. Díaz, and P. Gervás, "A semantic graph-based approach to biomedical summarisation," *Artificial intelligence in medicine*, vol. 53, no. 1, pp. 1–14, 2011.

[25] M. Moradi, "Cibs: A biomedical text summarizer using topic-based sentence clustering," *Journal of biomedical informatics*, vol. 88, pp. 53–61, 2018.

[26] A. Sarker, Y.-C. Yang, M. A. Al-Garadi, and A. Abbas, "A light-weight text summarization system for fast access to medical evidence," *Frontiers in Digital Health*, vol. 2, p. 45, 2020. doi: 10.3389/fdgth.2020.585559. [Online]. Available: https://www.frontiersin.org/article/10.3389/fdgth.2020.585559

[27] N. Rahman and B. Borah, "Improvement of query-based text summarization using word sense disambiguation," *Complex & Intelligent Systems*, pp. 1–11, 2019.