May 2014

CiteFinder: a System to Find and Rank Medical Citations

Seyed Soheil Moosavinasab
University of Wisconsin-Milwaukee

CITEFINDER: A SYSTEM TO FIND AND RANK MEDICAL CITATIONS

by

Soheil Moosavinasab

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

May 2014

ABSTRACT

CITEFINDER: A SYSTEM TO FIND AND RANK MEDICAL CITATIONS

by

Soheil Moosavinasab

The University of Wisconsin-Milwaukee, 2014

Under the Supervision of Dr. Rashmi Prasad

This thesis presents CiteFinder, a system to find relevant citations for clinicians' written content. Inclusion of citations for clinical information content makes the content more reliable through the provision of scientific articles as references, and enables clinicians to easily update their written content using new information. The proposed approach splits the content into sentences, identifies the sentences that need to be supported with citations by applying classification algorithms, and uses information retrieval and ranking techniques to extract and rank relevant citations from MEDLINE for any given sentence. Additionally, this system extracts snippets from the retrieved articles. We assessed our approach on 3,699 MEDLINE papers on the subject of "Heart Failure". We implemented multi-level and weight ranking algorithms to rank the citations. This study shows that using Journal priority and Study Design type significantly improves results obtained with the traditional approach of only using the text of articles, by approximately 63%. We also show that using the full-text, rather than just the abstract text, leads to extraction of higher quality snippets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I want to express my sincere gratitude to my advisor, Dr. Rashmi Prasad, for her patience, motivation, enthusiasm, and immense knowledge. She has always been encouraging me to pursue my ideas and goals in both academic and non-academic life. Her guidance helped me in all the time of research and writing of this thesis.

I would also like to thank my committee members, Professor Susan McRoy and Professor Jun Zhang for serving as my committee members.

My special appreciation and thanks also goes to Dr. Siddhartha Reddy Jonnalagadda, for offering me the summer internship opportunity at Mayo Clinic, where this project was started and built upon for my thesis. Without Dr. Jonnalagadda's supervision during the internship and his subsequent assistance and support, this work would have never been accomplished.

My sincere gratitude goes to Professor Ethan Munson, chair of the department of Computer Science for supporting me financially several times during my education at UWM. I also want to thank Professors Rashmi Prasad, Timothy B Patrick, Hong Yu, and my parents for their support during my master's education.

I thank my fellow lab mates in UWM Biomedical Data and Language Processing, especially Majid Rastegar-Mojarad, Feifan Liu, Shashank Agarwal, Brian Harrington, and Omid Ghiasvand for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two years. Very special thanks to Majid Rastegar-Mojarad for his comments and ideas in this project and being a supporting friend during all these years.

In addition, I would like to thank Professor Hong Yu, who helped me to get familiar with the Biomedical and Health related studies when I started my education at UWM.

Last but not the least, none of this could have happened without my family. I would like to thank my parents, brothers and sister in law for supporting me spiritually throughout my life.

Chapter 1:

Introduction

1-1 Motivation

Providing clinical information is a very important but also a very sensitive matter because it deals with people's health. A mistakenly advised drug or medical recommendation can irreparably harm the patient. Therefore, clinicians must be cautious about the sources that they take the information from to avoid misplaced decision-making.

Written materials are one of the primary sources of learning and transfer of knowledge. When we talk of scientific text, reliability of information is a critical issue. However, the more well-documented the information is, the more chance there is for the audience to trust the information and use it. One way to include documentation of evidence with information presentation is to include citations to documents that constitute the source of the information. Such documentation adds value to the evidence-based content that may otherwise suggest risk of jeopardizing patient health through misinformation.

Supporting textual information with documented evidence is crucial for resources such as UpToDate®[1] and PubMed Central®[2] that are meant to be used by clinicians and experts. On the other hand, in typical clinical resources such as WebMD®[3], Cleveland Clinic[4] and Medline Plus®[5] that are designed for non-expert users such as patients, families, and health care providers, users can benefit from the enriched text linked to the citations and get more detailed information about the medical information, including the underlying research or case studies that serve as the basis for the information. As an example, although Medline Plus does not include citations for sentences, it lists global references

[1] http://www.uptodate.com
[2] http://www.ncbi.nlm.nih.gov/pmc
[3] http://www.webmd.com/
[4] http://my.clevelandclinic.org/
[5] http://www.nlm.nih.gov/medlineplus/

for each article at the end of the text as a way to provide users the opportunity to gather more information if they so desire.

Furthermore, although the language and information provided in non-expert based healthcare websites are simple, rudimentary and easily readable by non-experts, the enriched version of the same content can encourage more groups of people such as both experts and non-experts to use them.

The importance of enriching the text with citations is so great that even collaborative sources such as Wikipedia ® support their context by adding references to the sentences [1].

The primary contribution of this thesis is a system, known as CiteFinder, to find citations for clinical text. We suggest sources that support clinicians' text, help them to verify their thoughts and findings, and also collect new ideas about the topics they are working on. A system such as CiteFinder will help in transforming the expert-based content paradigm (a paradigm not used by certain clinical knowledge systems such as UpToDate© [2], but relatively common among geographically close care providers [3]) to evidence-based medicine – the accepted paradigm [4]. Presenting potential citations to clinicians gives them the flexibility to easily author evidence-based guidance and FAQs for their peers. Furthermore, it would also be extremely helpful to add citations to information content that was written at a time when no supporting literature existed but that became available over time. Filling these evidential gaps in existing clinical text would be of enormous use to clinicians and the community at large. In other words, the use of the CiteFinder system

is not limited to finding citations for a new article being written – it can also assist in keeping existing articles updated.

1-2 System Architecture

Figure 1 illustrates the architecture of our CiteFinder system. The system is implemented as an Ajax based online tool that allows a user to submit some clinical text and receive back the top relevant citations for candidate sentences in the input text. CiteFinder consists of two major components: sentence selection and citation assignment. The citation assignment component itself contains four subcomponents: sentence expansion, citation extraction, citation ranking, and snippet generation, which are each in turn applied on the selected sentences. We briefly describe the function of each of the components below.

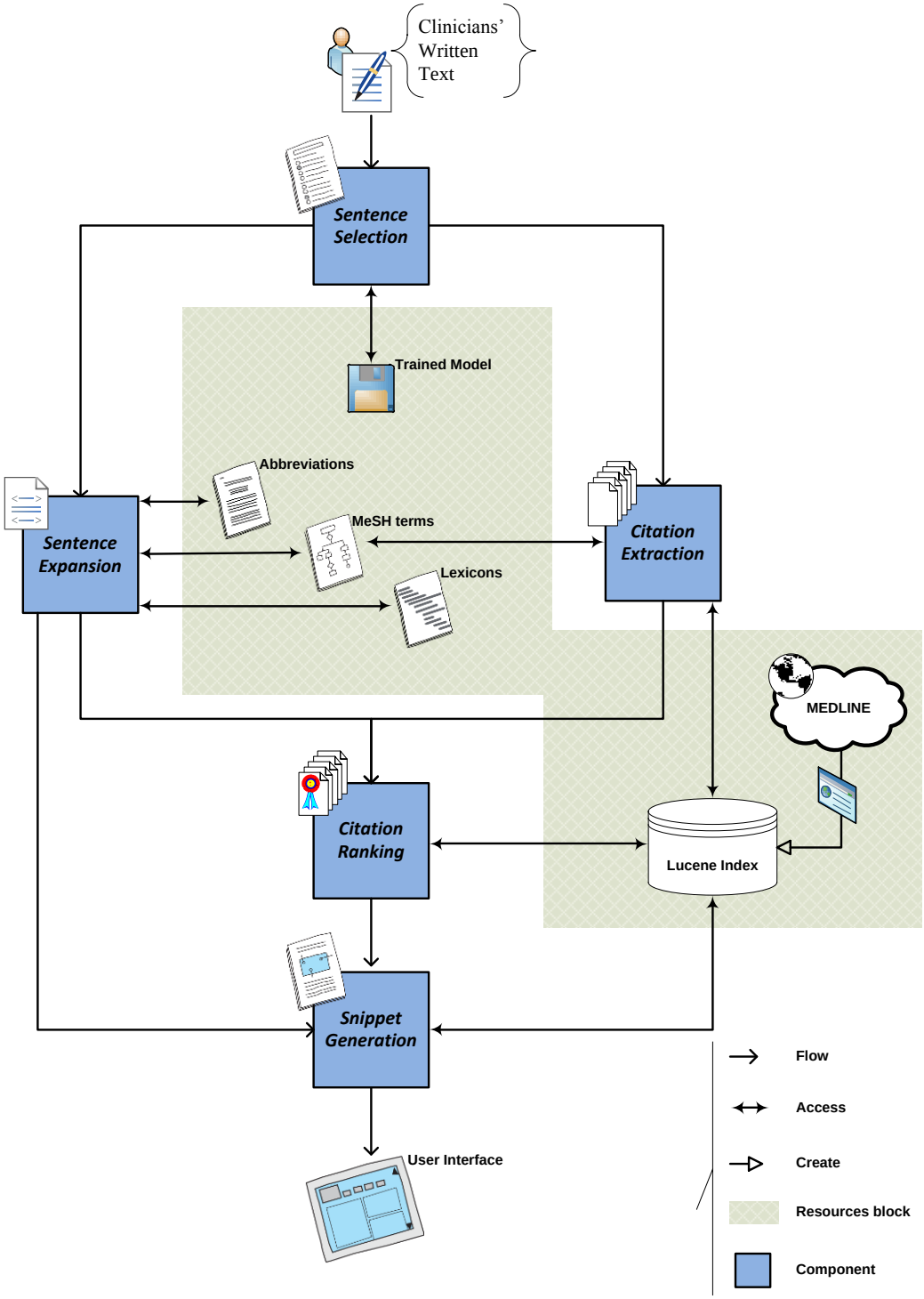


Figure 1: CiteFinder System Architecture.
The figure illustrates the sentence selection, sentence expansion, citation extraction, citation ranking, and snippet generation components and their integration with resources and the user-interface.

Sentence selection is the first component that processes the text submitted by the author. Since not all authored sentences are based on content drawn from some external source(s), the first task, therefore, is to identify the sentences whose content does depend on some external source(s) and for which external articles need to found and cited as documentation of evidence. For this, we use the Naïve Bayes binary classification algorithm, which classifies a sentence as either one needing a citation (+citation) or as one not needing a citation (-citation). Only +citation sentences are then passed on to the next component for citation assignment. An advantage of the sentence selection step in our approach is that the text resulting after the assignment of citations will be more readable and natural, with fewer false positives.

Sentences selected by the sentence selection component are sent for citation assignment, where each sentence is first expanded (Sentence expansion) by normalizing lexical variations, adding MeSH® (Medical Subject Headings) terms and abbreviation expansions to extend the word-based search scope for citations. Such expansion of the sentence allows the citation extraction component to retrieve additional relevant documents, thereby increasing recall, which in turn will enhance ranking during citation ranking, and eventually lead to the generation of more relevant snippets.

Sentence expansion is followed by citation extraction, which involves finding relevant citations for the candidate sentence, from our collection of MEDLINE® articles (see Section 1-3). To find relevant citations, MeSH terms are used. MeSH is a controlled vocabulary thesaurus created by the National Library of Medicine® (NLM) for the purpose of indexing and searching the content of MEDLINE articles. This enables

retrieval systems (e.g., PubMed[1]) to provide subject searching of the data.[2] The citation extraction component extracts MeSH terms from the sentence and searches for them among the MeSH terms of each indexed MEDLINE article. We retrieve articles that have at least one MeSH term in common with the sentence. We used the open-source Apache Lucene software[3] for indexing, search and retrieval.

Articles identified through MeSH search are then ranked by the citation ranking component, based on four measures: text search, MeSH search, journal prioritization, and epidemiological study design recognition [5]. We trained a system to give weights to each measure for full text MEDLINE articles and obtained the formula below to rank the articles based on the calculated score:

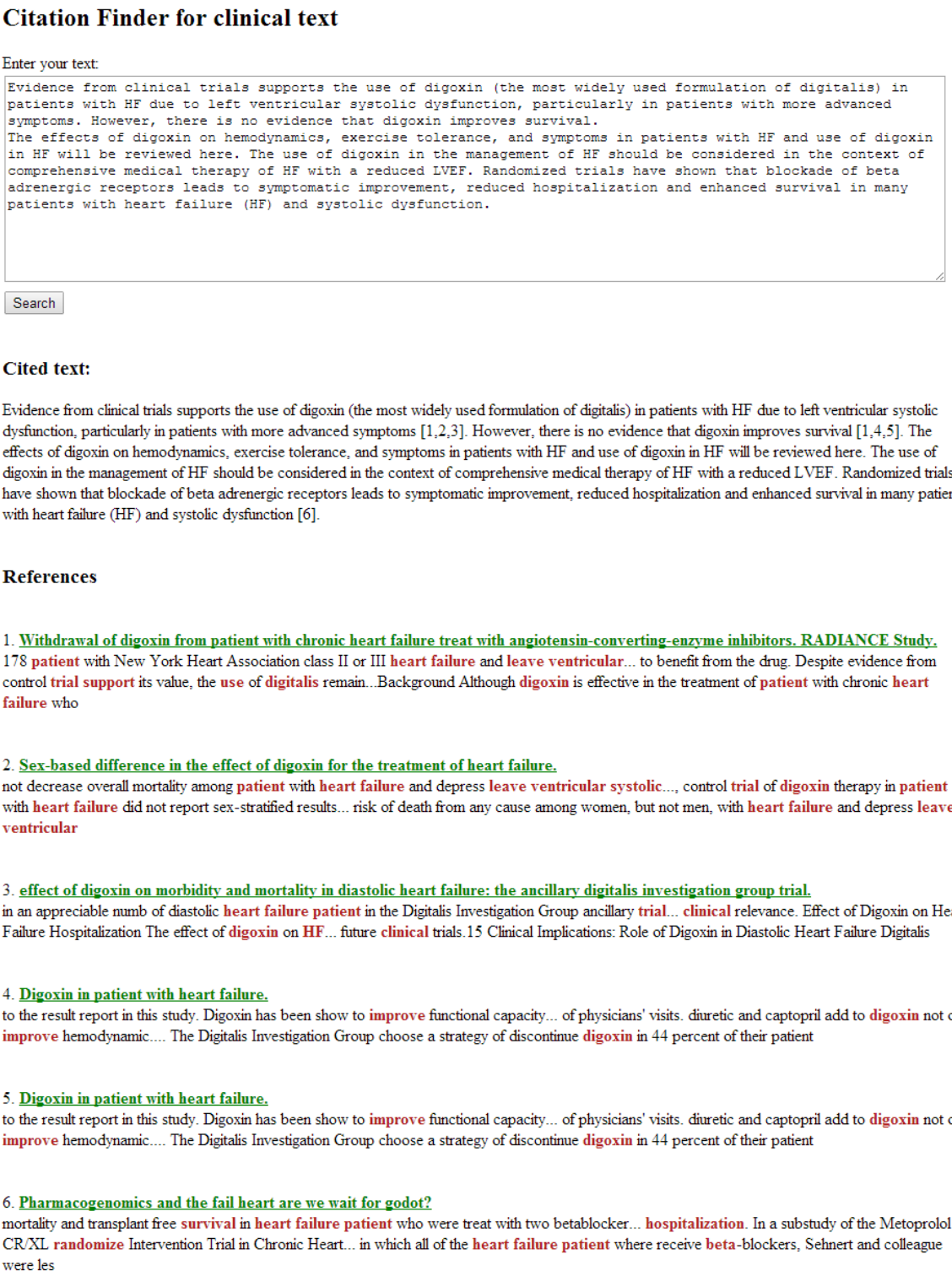
$$\text{Score} = (1 * \text{text search score}) + (0.45 * \text{Journal Prioritization score}) + (0.75 * \text{Study Design recognition score})$$

Citation ranking overcomes a common problem underlying most IR systems, namely that they retrieve an enormous number of articles. Ranking the retrieved documents spares users from combing through the less relevant results of the search. In addition, however, ranking of the retrieved results is crucial for a system such as ours, where the goal is not just to rank the articles but also to remove the less relevant articles. In our system, only the top 3 citations from the ranked retrieval results are returned to the user.

The final step of citation assignment involves producing snippets for the retrieved citations. A snippet is a small portion of the content that gives the user brief information about the retrieved document. In general, snippets can contain text summaries, image

[1] http://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.nlm.nih.gov/mesh/intro_retrieval.html
[3] https://lucene.apache.org

thumbnails, hyperlinks, dates, names, etc., in different fonts and formats. In our system, we generate snippets containing the title, the article URL, and a portion of the text of retrieved articles to show how the article provides good support for the sentence. Figure **2** shows a sample input text submitted by a user and cited output text along with list of references and snippets for them.



Citation Finder for clinical text

Enter your text:

```
Evidence from clinical trials supports the use of digoxin (the most widely used formulation of digitalis) in
patients with HF due to left ventricular systolic dysfunction, particularly in patients with more advanced
symptoms. However, there is no evidence that digoxin improves survival.
The effects of digoxin on hemodynamics, exercise tolerance, and symptoms in patients with HF and use of digoxin
in HF will be reviewed here. The use of digoxin in the management of HF should be considered in the context of
comprehensive medical therapy of HF with a reduced LVEF. Randomized trials have shown that blockade of beta
adrenergic receptors leads to symptomatic improvement, reduced hospitalization and enhanced survival in many
patients with heart failure (HF) and systolic dysfunction.
```

Search

Cited text:

Evidence from clinical trials supports the use of digoxin (the most widely used formulation of digitalis) in patients with HF due to left ventricular systolic dysfunction, particularly in patients with more advanced symptoms [1,2,3]. However, there is no evidence that digoxin improves survival [1,4,5]. The effects of digoxin on hemodynamics, exercise tolerance, and symptoms in patients with HF and use of digoxin in HF will be reviewed here. The use of digoxin in the management of HF should be considered in the context of comprehensive medical therapy of HF with a reduced LVEF. Randomized trials have shown that blockade of beta adrenergic receptors leads to symptomatic improvement, reduced hospitalization and enhanced survival in many patients with heart failure (HF) and systolic dysfunction [6].

References

1. **Withdrawal of digoxin from patient with chronic heart failure treat with angiotensin-converting-enzyme inhibitors. RADIANCE Study.**
178 **patient** with New York Heart Association class II or III **heart failure** and **leave ventricular**... to benefit from the drug. Despite evidence from control **trial support** its value, the **use** of **digitalis** remain...Background Although **digoxin** is effective in the treatment of **patient** with chronic **heart failure** who

2. **Sex-based difference in the effect of digoxin for the treatment of heart failure.**
not decrease overall mortality among **patient** with **heart failure** and depress **leave ventricular systolic**..., control **trial** of **digoxin** therapy in **patient** with **heart failure** did not report sex-stratified results... risk of death from any cause among women, but not men, with **heart failure** and depress **leave ventricular**

3. **effect of digoxin on morbidity and mortality in diastolic heart failure: the ancillary digitalis investigation group trial.**
in an appreciable numb of diastolic **heart failure patient** in the Digitalis Investigation Group ancillary **trial**... **clinical** relevance. Effect of Digoxin on Heart Failure Hospitalization The effect of **digoxin** on **HF**... future **clinical** trials.15 Clinical Implications: Role of Digoxin in Diastolic Heart Failure Digitalis

4. **Digoxin in patient with heart failure.**
to the result report in this study. Digoxin has been show to **improve** functional capacity... of physicians' visits. diuretic and captopril add to **digoxin** not onl **improve** hemodynamic.... The Digitalis Investigation Group choose a strategy of discontinue **digoxin** in 44 percent of their patient

5. **Digoxin in patient with heart failure.**
to the result report in this study. Digoxin has been show to **improve** functional capacity... of physicians' visits. diuretic and captopril add to **digoxin** not onl **improve** hemodynamic.... The Digitalis Investigation Group choose a strategy of discontinue **digoxin** in 44 percent of their patient

6. **Pharmacogenomics and the fail heart are we wait for godot?**
mortality and transplant free **survival** in **heart failure patient** who were treat with two betablocker... **hospitalization**. In a substudy of the Metoprolol CR/XL **randomize** Intervention Trial in Chronic Heart... in which all of the **heart failure patient** where receive **beta**-blockers, Sehnert and colleague were les

Figure 2: Demonstration of the user interface for CiteFinder system.
The system finds citations for some sentences that need citations to support them and creates a list of citations at the end.

Since the goal of the system is to show snippets to the users and provide them the opportunity to accept or reject the citations, articles with no extracted snippets are removed in this step. We analyzed abstract-based and full text-based production of snippets and evaluated the value of obtaining snippets not only from the abstracts, but also from the full text of the articles.

1-3 Evaluation Resources

Primary Gold Standard Corpus. For evaluating a system such as CiteFinder, what is needed is an existing resource of clinical informational text that (a) is written by human experts and (b) has citations provided (by human experts) for sentences, where needed as shown in Figure **3**.

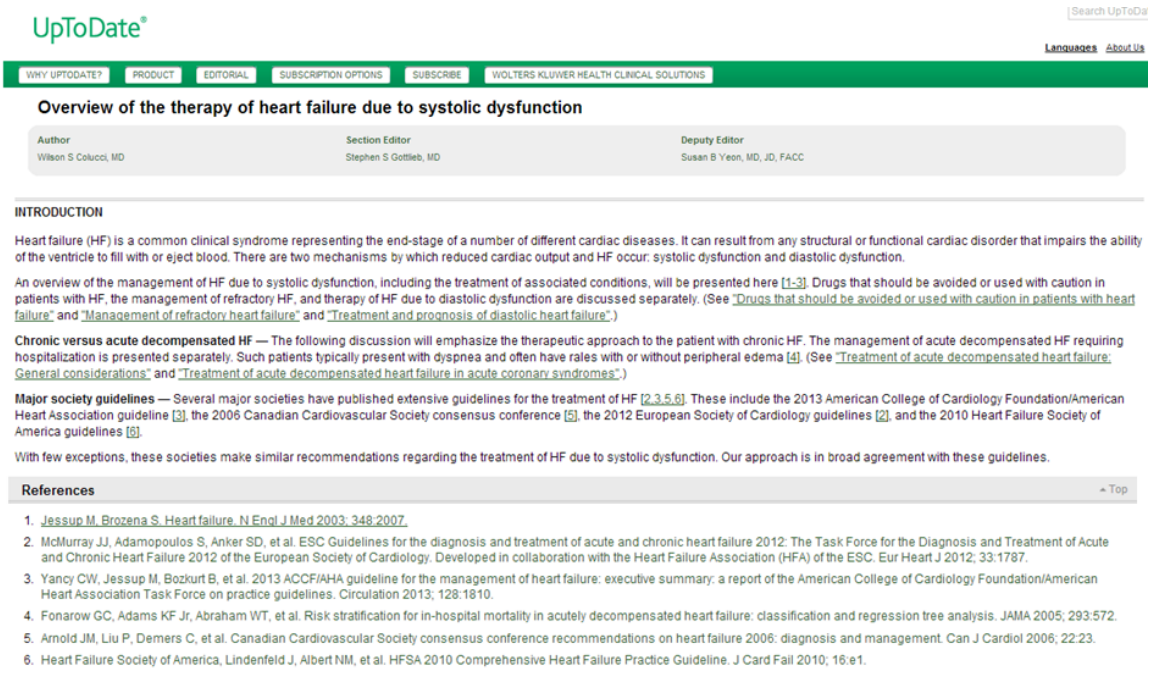


Figure 3: Sections of the UpToDate website.
The figure shows the text is supported with references for some sentences.

Sentences from such a resource can then be used as training and evaluation data for developing a method. In our work, we used UpToDate to create our primary gold standard corpus. UpToDate is an evidence-based, clinical decision support website which contains articles written by more than 5,100 physician authors, editors and peer reviewers[1]. Our reasons for choosing this resource as our gold standard were as follows. First, UpToDate is used by more than 700,000 clinicians from 158 countries, which shows that the information contained therein is regarded as highly reliable. Second, unlike many other clinical information sites, UpToDate includes citations in the articles at the sentence-level, where needed, instead of global citations for the article as a whole. This makes the data more conducive for our task evaluation. Finally, UpToDate articles and citations are updated on a regular basis by experts, thus reflecting the most recent status of the articles vis-à-vis current MEDLINE articles.

In this study, we selected "Heart Failure" topic as a representative of topics in the clinical domain. We obtained articles and journals, tested methods, and evaluated them in heart failure.

HTML files from UpToDate were obtained by downloading 150 articles retrieved with the query "heart failure". The content and citations were extracted from the HTML files by taking the text from the targeted sections using an HTML parser. Then a simple rule-based sentence splitter algorithm was applied to the text to delimit sentences, and two sets of sentences were created for our task: one containing sentences with one or more citations (i.e., sentences "needing citations", or +citation sentences) and the other containing sentences with no citations (i.e., sentences "not needing citations", or –citation

[1] http://www.uptodate.com/home/about-us

sentences). The final corpus created this way includes a total of 34321 sentences, with 7,757 sentences "needing citations", with citations to 11,793 articles, and 24,089 sentences "not needing citations".

Reference Article Collection. To find citations for the sentences in our corpus, we collected source articles comprising 3,699 articles on Congestive Heart Failure (CHF), from two major sources:

- 3,166 articles retrieved with the query "Congestive Heart Failure[MeSH Major Topic]" at PubMed Central
- 533 articles retrieved with the query "Congestive Heart Failure[MeSH Major Topic]" at PubMed on two top ranked journals for CHF topic. The articles are downloaded from the journal web sites directly:
    - JAMA the Journal of the American Medical Association
    - The New England Journal of Medicine

Since some articles in these two journals might also exist in PubMed Central, we removed duplicated articles and retained only one version of each. Also, documents for which only scanned versions were available were removed from the collection. In addition, we collected only those articles for which both the abstract and full text were available, in order to enable evaluation of our "full text-based" and "abstract-based" document retrieval algorithms. Both the abstract and the full-text of articles are indexed separately with Lucene to allow us to compare how our system performs in snippet generation task by searching over abstracts on the one hand and full-texts on the other.

1-4 Thesis Organization

The rest of this thesis is organized as follows. In chapter 2, we discuss related work on sentence classification (to identify sentences needing citations), citation finding systems, snippet generators, and ranking methods. Chapter 3 describes the sentence selection component, including the methods and evaluation results. Chapter 4 describes the citation assignment component, including its subcomponents for sentence expansion, citation extraction, citation ranking, and snippet generation. Methods and evaluation for this component are also discussed here. In chapter 5, we review the sources of errors and difficulties with the task and discuss conclusions and suggestions for future work.

Chapter 2:

Related Work

2-1 Introduction

In this chapter, we provide a review on studies relevant to this thesis. We report related studies and discuss our innovations for each of the sentence selection, sentence expansion, citation extraction, citation ranking, and snippet generation components.

2-2 Sentence Selection

Although there exists research on sentence selection for extraction-based summarization tasks [6]–[9], we are not aware of any research on sentence selection for citation finding. In work related to summarization, Yihong Gong et al [8] use traditional IR methods along with latent semantic analysis techniques for text summarization in the general domain. Ronald Brandow et all [6] also applied tf-idf approach on news publications to identify signature words and then used several factors to select sentences, including presence of signature words in the sentence, location of the sentence in the document, and other summarization related factors. Goldstein et all [7] also studied the task of summarizing news articles. They used statistical and linguistic features to score sentences. Statistical features include cosine similarity, tf-idf weights, pseudo-relevance feedback [10], query-expansion, and methods that eliminate text-span redundancy such as Maximal Marginal Relevance. Linguistic features are quotations, honorifics, and thematic phrases in this study. Daniel McDonald et all [9] implemented TXTRACTOR tool which generates summaries that contain user-defined number of sentences. Unlike most of the sentence selection tasks in summarization that rank sentences to select them [6]–[8], we trained a binary classifier using machine learning techniques to identify “needing citations” or “not needing citations” sentences.

Sentiment analysis is another task that uses sentence classification approach to classify opinions in a sentence. Peter D. Turney [11] studied classification of reviews on the internet as recommended (thumbs up) or not recommended (thumbs down). He calculates the semantic orientation of a phrase by considering the value of mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". If the average semantic orientation of all the phrases is positive, the review will be classified as positive. Bo Pang et all [12] studied the same task on multi-class text classification instead of binary classification. In another study [13], they used machine learning algorithms and combined the sentiment analysis and summarization tasks to obtain sentences with specific sentiments in documents.

Classifying the rhetorical function of a sentence is another task that has been studied in some depth. Simone et al [14] at first identify all the sentences that contain any rhetorical role. Then they classify identified sentences according to their rhetorical role.

Our work is the first for designing a sentence selection algorithm to find sentences which need citations, specifically in the clinical domain. Although most of the systems apply sentence selection algorithms in generic or news-related field [6]–[9], Larry et al [15] worked on the sentence classification task in the clinical domain to classify sentences functionally in terms of the major functional sections of articles, namely, Introduction, Method, Result, and Conclusion; but not on citation needing classification yet.

2-3 Citation Extraction and Ranking

Publication finding in digital libraries and on the web has been investigated to recommend relevant papers to researchers [16]–[20]. There are also studies on information retrieval in the medical domain. For example, Plaza and Diaz [21] proposed a method to query similar Electronic Health Records using UMLS concepts. Hersh and Hickam [22] studied the effectiveness of electronic information retrieval systems for physicians. Lu [23] investigated web tools for searches in the biomedical literature. Bachmann et al [24] proposed and validated search strategies used to identify diagnostic articles recorded on MEDLINE, with special emphasis on precision. Bernstam et al [25] studied how citation-based algorithms that are developed to extract relevant and important citations for the World Wide Web are useful in the biomedical literature domain. They compared eight citation algorithms, including simple PubMed queries, clinical queries, citation counts, journal impact factors, etc. Their research concluded that these citation-based algorithms are useful in the domain of biomedical literature. Lin et al [26] extracted relevant MEDLINE citations and ranked them based on several ranking methods, including citation counts per year and journal impact factors. Darmoni et al [27] used MeSH concepts for indexing and information retrieval. Some studies have also been conducted on query expansion using MeSH terms in PubMed. Lu et al [28] analyzed the effect of using MeSH terms in a PubMed automatic search. In the current study, we also used MeSH concepts to find relevant citations.

Simone et al [29] designed a machine learning framework to automatically classify function of a citations in 12 categories such as weakness, neutrality, and contrast or comparison, and agreement of the work. In other study [30] they implemented a multi classifier for identifying scientific attributions to improve discourse classification task.

They identify whose work is being talked about in the discourse. Current-Paper and No-Specific-Paper are two referents that they identify.

Some work has been done to retrieve articles form MEDLINE. Siddhartha et al [31] studied the information retrieval and semantic information extraction techniques to extract topic-relevant sentences from MEDLINE abstracts for clinician's queries. Sneiderman et al [32] introduced three knowledge based systems that help the clinicians to find answers for questions in MEDLINE. DingCheng et al [33] implemented a three step system for assigning references to expert-written content. They only studied the task on MEDLINE abstracts, not full text articles as our system does. They also have not automated the evaluation of the system but rather, selected the gold standard manually. In addition, adding text and MeSH search in our study shows improvements in the ranking results (the median rank on their study was 45 while it is 4 in our system).

2-4 Snippet generation

Many studies have been done on snippet generation in information retrieval in the web environment. Qing Li et all [34] discussed statistical language models to identify certain personalized patterns for snippet extraction. Other studies are available that discuss automatic snippet extraction techniques along with storage and efficiency optimizations [35]–[38]. In this study, we simply created a query of words, MeSH terms and expanded abbreviations and used Lucene's proximity search algorithm to find snippets.

Chapter 3:

Sentence Selection

3-1: Motivation

In any given clinical informational text, there will be some sentences that need citations to support the text while other sentences don't. Our study on expert-written articles in UpToDate shows that on average, 19.96% of the sentences in an article need at least one citation. The process of sentence selection is useful for citation finding as it filters out sentences that don't need citations. Sentences classified as "needing citations" by the sentence selection component will be used as input to the citation assignment component, leading to faster, more accurate, and more realistic results.

To understand and characterize the reasons for *why a given sentence may not need citations*, we carried out a study of 4 articles (total of 1,250 sentences) from our corpus. While not necessarily exhaustive, the characterization is intended to illustrate the range of sentence types that can be clearly identified as not needing citations. In addition, the study enabled us to develop features for the classification task. In the categorization below, sentence types that may plausibly require citations over time with the generation of new literature are marked as [inconstant] while others are marked as [constant]. One of the salient observations we made from the study was that certain word or phrase patterns are highly indicative of the sentence type in general. For each example sentence provided below for the sentence types, we have highlighted such indicative words and phrases in boldface.

- *Text Organization [constant]*: These include sentences that are used to structure the text to make it coherent for the reader. Since they are not primarily intended to present information, such sentences typically don't require any citations.

Examples:

- "**The following discussion** will emphasize the therapeutic approach to the patient with chronic HF."
- "Two major findings **were noted**:"
- "These issues **are discussed in detail** separately."

- *Opinions [constant]*: These sentences express findings or thoughts of the authors of the text. As such, we don't expect such sentences to contain citations to other sources.

Examples:

- **We prefer** to avoid nonsteroidal anti-inflammatory medications (NSAIDs) in elderly trauma patients.
- "**We** also **believe** it is reasonable to use a pulse of 90 or above as the threshold defining tachycardia, which may be a sign of hemorrhage or significant injury warranting careful investigation."
- "**We believe** that clinicians should choose one of the beta blockers of proven benefit (including reduction in all-cause mortality) in randomized trials (ie, carvedilol, extended-release metoprolol succinate, or bisoprolol)."

- *Internal references [constant]*: These include sentences that refer to statements, sections, figures, tables, etc., within the article, and therefore don't need to cite external sources.

Examples:

  - "(**See** 'Diuretics' **below**.)"
  - These studies **are listed in the text**.
  - "Measures of this property include (**figure 7**):"

- *No study exists [inconstant]*: These include sentences where the authors of the article explicitly state that there (currently) exists no work (or not enough work) on the topic in question.

Examples:

  - **There are not enough data** at present to recommend cardiac rehabilitation for patients with advanced HF.
  - **There is no** clear definition of geriatric trauma; in this topic, we include patients over 65 years.
  - **There is no evidence to support** firm guidelines, and the frequency and intensity of reevaluation will vary depending upon the baseline health of the patient, the clinical scenario, and available resources.

- *Questions [constant]*: These include questions posed by authors and, therefore, don't constitute information from external sources.

Examples:

- **What** medications is the patient taking (eg, anticoagulant, antiplatelet, beta blocker, calcium channel blocker)**?**
- **What** was the patient's baseline level of motor and cognitive function prior to the traumatic event**?**
- **What** underlying illnesses does the patient have (eg, cardiovascular or renal disease, diabetes)**?**

- *Suggestions [constant]:* These include advice and suggestions given by authors. Therefore the sentences represent newly generated content and do not need citations.

Examples:

- We **suggest** that trauma patients over the age of 70 be evaluated at a trauma center whenever possible, regardless of the mechanism of injury.
- We **encourage** you to print or e-mail these topics to your patients.
- For patients with current or prior HF and an LVEF <=40 percent, we **recommend** therapy with a beta blocker (Grade 1A).

- *Future work [inconstant]:* These include sentences describing ideas to expand the current study in the future. Such sentences indicate that the authors have themselves searched the state of the art publications and have not found any relevant articles to demonstrate the work. Therefore there is no need to search for citations at

the current time, although relevant supporting articles may appear in the future literature as and when the corresponding studies are conducted and published.

Examples:

- **Further study** is needed to better define the potential role of inflammation and associated alterations in DHF.

- *Findings [constant]:* These include authors' new observations, findings, and reported information. Since such information is generated exclusively by the authors of the article, citations for such sentences are not required.

Examples:

- **We obtain** basic laboratory studies in elder trauma patients with known or at significant risk for major injuries.

- Given these considerations, **we** start with a low dose of an ACE inhibitor (eg, lisinopril 5 mg/day), increase to a moderate dose (eg, lisinopril 15 to 20 mg/day) at one to two week intervals, and then begin a beta blocker, gradually increasing toward the target dose or, if this cannot be achieved, the highest tolerated dose.

- **We believe** that clinicians should choose one of the beta blockers of proven benefit (including reduction in all-cause mortality) in randomized trials (ie, carvedilol, extended-release metoprolol succinate, or bisoprolol).

3-2 Gold Standard

We trained a binary classifier to identify which sentences need citations and which don't. For training and evaluation of the classifier, we chose a subset of the gold standard data, with equal numbers of positive and negative instances, in order to avoid classification bias due to the highly unbalanced distribution of the data:

- +citations (positive instances): Randomly selected 4000 (out of 7,757) sentences from the "needing citations" set of sentences in primary gold standard corpus
- -citations (negative instances): Randomly selected 4000 (out of 24,089) sentences from the "not needing citations" set of sentences in primary gold standard corpus

3-3 Methods

As the first step, we split the given text into sentences using a simple rule-based java sentence splitting algorithm. Splitting the text into sentences gives us the ability to treat each sentence as an independent query for sentence selection and citation assignment. Each sentence is then pre-processed (Section 3-3-1) to remove noise and enforce normalization. Machine learning algorithms are then applied (Section 3-3-2) for training the classifier. We experimented with different algorithms and explored various features and feature combinations. As shown in our results in Section 3-4, the best results for this task are obtained with a Naïve Bayes classifier, with the best feature being the top 6000 of the most frequent words.

3-3-1 Pre-processing

Machine learning is highly dependent on consistency in the input data, for which removing noise and enforcing normalization, where possible, is an important step. Each sentence in the gold standard data was pre-processed for the following:

- *Character normalization*: Unconventional characters and symbols are converted to machine readable forms, specifically:
  - ± → +-
  - — → -
  - ≤ → <=
  - ≥ → >=
  - β → b
- *Number normalization*: Since numbers don't contribute to discrimination for this particular sentence classification task, all numbers are normalized by substituting them with the same number ("99999").

3-3-2 Classification

We used the Weka© [39] 3.6 workbench to apply classification algorithms on the gold standard. Weka is a freely available suite of machine learning tools for data mining and data analysis tasks. Weka accepts Attribute-Relation File Format (ARFF) files as input, with each instance described as a set of attributes. We explored combinations of different features as attributes, values for which were automatically extracted from the gold standard sentences. Training and testing was done using 5-fold cross validation.

Features

Here we describe the features we used for the sentence classification task, including how they were obtained:

- **All words:** Words were obtained after applying tokenization. Tokenization was done by simply using the white space as a delimiter. A list of all words occurring in the corpus was used to create features indicating whether a word in the list occurred in the sentence or not.

- **Stemmed words:** Stemmed versions of words were obtained with the Porter stemmer [40], and a similar process as above was used to create features for the stemmed words.

- **Words with stop words removed:** Stop words are very commonly occurring function words (e.g., *the*, *a*, *an*, *of*, *for*, etc.) that are often removed in classification tasks when words are used as features. Their removal is found to improve system speed, efficiency and performance in many cases. We used a list of 119 stop words obtained from *text fixer*[1]. The same process as above was applied to create features after stop words were removed from the sentences.

- **MeSH terms:** We wanted to explore if certain MeSH terms could be more indicative than others for whether the sentence needs a citation or not. We extracted MeSH terms in the sentences and created a list of all the MeSH terms in the sets. Existence or lack of existence of each MeSH term then defines this

[1] http://www.textfixer.com/resources/common-english-words.txt

feature. The details about what is MeSH term and how we extracted them are discussed in Chapter 4, Section 4-2.

- **Number of words in a sentence:** The number of words occurring in the sentence was used as a feature, although, since the average number of words in +citation sentences and –citation sentences did not show much of a difference, we hypothesized the contribution of this feature to be weak at best.

- **Length of the sentence:** The length of the sentence reflects the total number of characters in the sentence, but again, for the same reason as above, we deemed the contribution of this feature to be weak.

- **Patterns:** Based on a study of a portion of the data, we collected patterns that recurred in one or the other type of sentences. For example, sentences starting with the "(see" phrase were often found to be –citation sentences.

- **Word N-grams:** We explored the impact of word N-grams (n = 2 to 5) as a feature. We used N-grams and combinations of N-grams in two ways, one using all words, and another using words after removing stop words.

- **Number of semantic role relations:** A semantic role is the underlying relationship that an entity has with the main verb in a sentence. SemRep [41] is a Natural Language Processing tool designed by National Library of Medicine that extracts semantic predications (subject-relation-object triples) from biomedical text using underspecified syntactic analysis and structured domain knowledge from the UMLS. We used SemRep to determine the number of semantic relations

appearing in a sentence and used this as a feature, to explore the idea that the more complex a sentence, the more likely it may be to require a citation.

- **POS tags:** We parsed all sentences using the Stanford Parser 3.2.0 [42] and extracted word POS tags, using them as a feature. For this feature, words were used in their bare form and stop words were not removed.

- **N-grams of POS tags:** We used the same method used for word N-grams to collect POS N-grams.

Feature selection

After generating the features, we tested using only the features which appeared the most. We selected top 300, 100, 2000, and 6000 features in each test to study the improvements with feature selection.

Feature selection in helpful in the classification system for several reasons:

1. *Improves the results*: Feature selection usually leads to performance improvements by selecting valuable features and removing noisy features.

2. *Avoids overfitting*: Overfitting occurs when the classifier makes an incorrect generalization because it also considers noise in the data instead of just the essential features. Feature selection increases the chance of removing noise and therefore reduces the possibility of overfitting.

3. *Makes using of all classifiers feasible*: If the size of the training data is too large, it may become impractical to run the classifier because of memory limitations on

the computer. Feature selection reduces the size of the training data by choosing only the important features so that the data can be processed by the system.

4. *Increases the speed*: Even if the system is able to handle the training data, feature selection will reduce the training time by reducing the number of features.

Algorithms

We conducted experiments with seven different classification algorithms in Weka to determine which would work best for the sentence selection task:

- Naïve Bayes
- SVM
- Decision Stump
- Random Forest
- MultiBoostAB
- Conjunctive Rule

In addition to the above algorithms, we considered using SMO, KStar and Decision Table too, but decided against them eventually because they were too slow. The results in Table 1 show that Naïve Bayes, with an average F score of 0.709, is the best algorithm for the sentence selection task.

Naïve Bayes classifier is a supervised learning algorithm adapted from Thomas Bayes' theorem. In this theory, features are assumed to be independent of each other such that the presence or absence of a particular feature in a class is assumed to be unrelated to the

presence or absence of any other feature in the class. The advantage of Naïve Bayes classifier is its simplicity, computational efficiency, and reasonable classification performance.

The Bayes' rule indicates that the probability of a document "*d*" being in a class "*c*" is:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Where

P(c|d) is the posterior probability of class *c* to be chosen for the given document.

P(c) is the prior probability of the class.

P(d|c) is the probability of document *d* to belong in class *c*.

P(d) is the prior probability of the document.

So we calculate the Maximum Posteriori Hypothesis (MAP) for each class using the following formula:

$$c_{MAP} \equiv argmax\ \frac{P(d|c)P(c)}{P(d)}\ for\ each\ c\ \in\ C$$

We remove *P(d)* from the equation because its value is constant.

$$c_{MAP} \equiv argmax\ P(d|c)P(c)\ for\ each\ c\ \in\ C$$

If we imagine document "*d*" consists of terms such as $\{x_1, x_2, \ldots, x_n\}$, the equation can be replaced by:

$$c_{MAP} \equiv argmax\ P(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n|c)P(c)\ for\ each\ c\ \in\ C$$

Now, if we apply the Naïve Bayes independence assumption, we can imagine that the probability of observing the conjunction of words is equal to the product of the individual probabilities:

$$P((\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_n|c) = P(\mathrm{x}_1|c) * P(\mathrm{x}_2|c) * \dots * P(\mathrm{x}_n|c)$$

Therefore we can use the formula below:

$$c_{MAP} \equiv argmax\, P(c) \prod_{i=1}^{n} P(\mathrm{x}_i|c) \;\; for\, each\, c \;\in\; C$$

Where,

$$P(\mathrm{x}_i|c) = \frac{Number\ of\ times\ word\ x_i\ occures\ in\ the\ class\ c}{Number\ of\ words\ in\ class\ c}$$

3-4 Results

As discussed in the methods section, we trained and tested on the gold standard sentences using Weka classifiers and got the best results with the Naïve Bayes algorithm and the "bag of words" feature.

3-4-1 Evaluation metrics

We use standard Precision, Recall and F-1 measures to evaluate the results. Precision (also known as positive predictive value) is the proportion of returned instances that are relevant. High precision means that the system identified more relevant instances than irrelevant [43].

$$Precision = \frac{\{relevant\ instances\} \cap \{retrieved\ instances\}}{\{retrieved\ instances\}}$$

In other words, we can calculate precision by True Positive (TP), and False Positive (FP) values:

$$Precision = \frac{TP}{TP + FP}$$

Using only precision in the evaluation will reveal how relevant the retrieved results are, but it will not give us any information about how comprehensive the results are. For example, if the system only returns one TP instance, the precision will be maximum, but it ignored the many more instances that the system should have correctly returned (i.e., the False Negatives). We use recall in our evaluation to account for such errors.

Recall (also known as sensitivity) is the proportion of relevant instances that are returned. High recall means that most of the relevant instances are returned by the algorithm [43].

$$Recall = \frac{\{relevant\ instances\} \cap \{retrieved\ instances\}}{\{relevant\ instances\}}$$

Recall can be calculated as the portion of True Positives and False Negatives (FN) that are TP.

$$Recall = \frac{TP}{TP + FN}$$

Considering only recall in the evaluation can lead to misleading results as well. If the system returns all the instances regardless of their relevance, the recall will be maximum. This is because the recall metric does not consider false positive cases in the evaluation.

A metric that balances Precision and Recall is more desirable. For this, we use the F-1 score which provides the harmonic mean of Precision and Recall. The value of the F-1 score (or simply f score) lies between 0 and 1, and is calculated as follows:

$$\mathrm{F1} = 2 * \frac{precision * recall}{precision + recall}$$

We used the 5-fold cross-validation evaluation method to partition the data, and use Macro-average rather than Micro-average for the F score. Macro-average F scores are calculated by first calculating the F score for each class and then taking the average of these. On the other hand, Micro-average scores aggregate the TP, FP, and FN for all the classes and then calculate the precision, recall, and F-1 scores. Because the F-1 metric does not consider true negatives (TN) and its score is mostly affected by true positives, large classes control the result more than small classes in micro-averaging. Macro-averaging is a better evaluation metric in our system because it gives equal weight to every class (class-pivoted measure) while micro-averaging gives equal weight to each sentence classification decision (it is called a document-pivoted measure) [43]–[45].

3-4-2 Algorithm Selection

We chose 5 candidate features to study the impact of different algorithms on our gold standard and then chose Naïve Bayes as the best algorithm for sentence selection task, based on the results. The 5 specific features were selected because we considered them to be most representative of all the features and because they are also generally found to be used in other classification tasks. Table 1 shows the results obtained with different algorithms. Precision, recall and F-1 measures are listed in three rows, respectively, in each cell of the table. This experiment for comparing classification algorithms was

performed using only the top 1000 features. The impact of feature selection itself is discussed later in section 3-4-3.

Table 1: Results for sentence classification with different algorithms and 1000 top features

	1	5	9	9	14	Average
Features / Algorithms	All Words	MeSH terms	1 + Patterns	Bigram of words	POS	
Naïve Bayes	0.753 0.749 0.748	0.604 0.603 0.603	0.752 0.749 0.748	0.724 0.711 0.707	0.75 0.744 0.743	0.716 0.711 0.709
LibSVM	0.725 0.708 0.702	0.605 0.605 0.605	0.706 0.676 0.664	0.703 0.658 0.637	0.724 0.707 0.701	0.692 0.67 0.661
Decision Stump	0.758 0.579 0.491	0.655 0.577 0.516	0.759 0.576 0.484	0.765 0.574 0.48	0.76 0.577 0.487	0.739 0.576 0.491
Random Forest	0.731 0.729 0.729	0.589 0.589 0.588	0.732 0.731 0.73	0.701 0.699 0.698	0.729 0.727 0.726	0.696 0.695 0.694
MultiBoostAB	0.758 0.579 0.491	0.628 0.628 0.628	0.759 0.579 0.49	0.765 0.574 0.48	0.76 0.577 0.487	0.734 0.587 0.515
Conjunctive Rule	0.758 0.579 0.491	0.622 0.582 0.544	0.759 0.576 0.484	0.765 0.574 0.48	0.76 0.577 0.487	0.732 0.577 0.497

Since Naïve Bayes clearly gives us the best results on the F score, all further experiments to explore the impact of various features are conducted with this algorithm.

3-4-3 Features

Table 2 shows the impact of different feature types (Section 3-3-2) as well as the impact of selecting different top N features on the sentence classification task.

Table 2: Naive Bayes classification with different features

	Features	300	1000	2000	6000
1	All Words	0.729 0.724 0.722	0.753 0.749 0.748	0.768 0.765 0.764	0.783 0.78 0.779
2	1 + Stemming	0.731 0.727 0.726	0.756 0.752 0.751	0.769 0.766 0.765	0.779 0.775 0.774
3	1 + remove stop words	0.727 0.721 0.719	0.752 0.748 0.747	0.769 0.766 0.765	0.782 0.779 0.778
4	1 + 2 + 3	0.731 0.727 0.726	0.756 0.752 0.751	0.769 0.766 0.765	0.779 0.775 0.774
5	MeSH terms	0.604 0.604 0.603	0.604 0.603 0.603	0.605 0.605 0.604	0.606 0.605 0.605
6	1 + number of words	0.722 0.716 0.714	0.751 0.746 0.744	0.77 0.766 0.765	0.78 0.776 0.776
7	1 + Length of the sentence	0.724 0.718 0.716	0.752 0.747 0.746	0.768 0.765 0.764	0.78 0.776 0.776
8	1 + Patterns	0.726 0.723 0.721	0.752 0.749 0.748	0.768 0.766 0.766	0.78 0.779 0.778
9	Bigram of words	0.707 0.692 0.686	0.724 0.711 0.707	0.739 0.727 0.723	0.757 0.748 0.745
10	Trigram of words	0.687 0.65 0.633	0.715 0.686 0.675	0.721 0.698 0.689	0.729 0.709 0.703
11	1 + 9 + 10	0.722 0.712 0.709	0.737 0.726 0.722	0.748 0.736 0.733	0.764 0.753 0.751
12	1 + 3 + 9 + 10	0.724 0.712 0.708	0.743 0.73 0.727	0.755 0.744 0.741	0.774 0.764 0.762

13	3 + 9	0.713 0.689 0.68	0.723 0.705 0.698	0.731 0.717 0.713	0.75 0.736 0.732
14	POS	0.726 0.721 0.72	0.75 0.744 0.743	0.767 0.762 0.761	0.776 0.772 0.771
15	1 + Relation count	0.728 0.723 0.722	0.753 0.749 0.748	0.767 0.763 0.763	0.782 0.779 0.778

3-5 Discussion

In this section, we discuss sentence selection component's methods, features, and evaluation resources. We present ideas of improving the performance by analysis the results we obtained in this task.

Features contributions

As we discussed in the motivation section, we explored the gold standard sets and observed 8 different situations in text when a given sentence may not need citations. This task helped us to acquire list of characteristics of sentences that do not need citations to be supported. As we highlighted observed patterns and key words in that section, most of the "not needing citations" cases that we detected are very key word sensitive. Sentences with words such as *following, believe, see, below, there is/are no/not, what, suggest, further, and believe* are very likely to be identified as "not needing citations". Results we obtained by selecting different combinations of feature are shown in Table 2. Since the patterns we observed show that general words have an important role in identifying sentence class in this task, weak contribution of structure-based features such as POS

tags, and domain specific words such as MeSH terms was expected. Bag of words on the other hand showed the best performance. Adding N-grams did not improve the results too because sensitive words in this task are not necessarily followed in a single sequence.

Algorithm selection

Different kinds of classifiers should be always considered for a comparative study over a given dataset. We tried six different classification algorithms and selected Naïve Bayes as the best one in the sentence selection for citation finding task. Our study on different types of sentences that do not need citations showed that the classification task is mostly word based. Independence assumption of Naïve Bayes algorithm fulfills this requirement and contributes the best in the task. We think that the independence assumption is the reason we obtained better results over SVM which is also a word sensitive algorithm.

Random forest is a classification method that operates by constructing a multiple of decision trees at training time. We receive the second best result in the classification task using random forest algorithm. Decision stump on the other hand operates only on one-l3v3l decision tree. Although decision stump is faster to run in our experiment, random forest has a better performance in results because of operating on multiple decision trees.

MultiBoosting develops a classifier in the form of a committee of subsidiary classifiers. Individual output of the committee classifiers combine to create a single classification from the committee as a whole, often performed by majority vote. MultiBoostAB implements cascade classifier to obtain the final class by voting. This algorithm is known to be sensitive to noisy data.

Although rule based algorithms cannot identify many positive instances (low recall), they are relatively precise (high precision). Conjunctive rule is a rule based classifier we used in sentence selection task. Since our study on "not needing citations" sentence types showed sentences are mostly identified by words but not rules, we were expecting the minimum contribution from conjunctive rule algorithm.

Baseline: Random choice algorithm

0.779 is the best f score obtained using Naïve Bayes algorithm and 6000 most frequent words as features. In comparison, we calculated the scores with random choice of the class, which is when we randomly assign sentences to one of the two designated classes. The results for the random choice evaluation are as follows:

Accuracy = (4000) / (4000 + 4000) = 0.5

P1 = (2000) / (2000 + 2000) = 0.5

R1 = (2000) / (2000 + 2000) = 0.5

F1 = 2 * (0.5 * 0.5) / (0.5 + 0.5) = 0.5 / 1 = 0.5

Comparing the results for random choice method with the binary supervised classification shows that the Naïve Bayes classification improves the classification results more than 50%.

Noisy gold standard

In this task, we experimented with many features and algorithms to obtain the best result. However, adding more features mostly didn't result in significant improvements. One reason is that the data sets we used to train the system are very noisy. In particular, we

observed that while some sentences should have included a citation, the author nevertheless did not include it. This could have been due to lack of time, a different point of view on whether the sentence requires a citation, or a failure in finding a citation. For the purpose of our task, what is worth noting is that the dataset used here is likely to have this degree of noise, because of which addition of features that seemed intuitively plausible did not improve performance in the end. As future work, we believe that the task may benefit from methods such as crowd sourcing to collect gold standard data that is more reliably marked for citations.

Number of semantic role relations as a feature

We added "number of semantic relations in a sentence" as a feature following the intuition that the more relations there are in a sentence, the more likely it may be for requiring citations. Although this feature did not help to improve the results any further over the words, we think that counting specific relations such as "TREAT_OF" instead of all occurring relations may be helpful for discrimination.

Synonymous expansion and grouping

Our study on 4 articles (1,250 sentences) and also the results we obtained from the classification task show that the task of selecting sentences that need citations to be supported is very word sensitive. In the future, we can use tools such as WordNet[1] to expand and group synonymous words. This approach will improve the classification results by considering the semantic of words. For example, *suggestion* sentences are one of the types that we observed don't need citations. Some of the words that indicate this sentence type are "suggest" and "recommend". By grouping these words together, we

[1] http://wordnet.princeton.edu/

will have a stronger learning algorithm that is not mislead by same semantic but different characterized words.

Discourse analysis

In this study, we are not gaining any information from the relations between the sentences. We can have stronger sentence selection methods by considering the relationships between sentences using discourse analysis.

Chapter 4:

Citation Assignment

In this chapter, we describe the citation assignment component of CiteFinder. Sentences received from the sentence selection component are first expanded and then submitted for citation extraction, which involves finding relevant citations for the candidate sentence using MeSH terms. Articles identified through this MeSH-based search are then ranked based on four measures: *text search*, *MeSH search*, *journal prioritization*, and epidemiological *study design recognition*. The final step involves producing snippets for the retrieved citations, where a snippet contains the title, the article URL, and a portion of the text of retrieved articles to show how the article provides good support for the sentence. Snippets are obtained not only from the abstracts, but also from the full text of articles. In what follows, we describe the gold standard used to evaluate the task and also our methods for each of these subtasks in detail and present results from our experiments.

4-1 Gold Standard

Since the sentences in the "needing citations" set have at least one citation, we created a gold standard using this set. Although all the sentences seem to be a good candidate for the system, we had to filter out some sentences because of the following reasons:

1. *No PMID available [420 sentences removed]*: There are a few sentences in UpToDate, where, although the authors mentioned a citation or citations for the sentences, there is no reference provided on the webpage for the citation. These citations may be updated in the future with the correct reference provided, but we removed them for now in our evaluation.

2. *No full-text available in our index files [6,844 sentences removed]*: We obtained the "needing citations" sentences by crawling the articles from

UpToDate and extracting the sentences with at least one citation. The advantage of using UpToDate is that all the citations in this web site are from MEDLINE, which is appropriate for our system because the indexed article collection is obtained from the same source. On the other hand, since we are interested in using the full text of articles in addition to the abstracts, we had to remove the articles for which the full text was not available.

3. *Less than 15 words [87 sentences removed]*: Since our algorithms for both sentence selection and citation assignment are word sensitive, we only selected sentences which had at least 15 words. For example, partial sentences such as "The main changes are" are removed.

4. *Less than 1 MeSH term [2 sentences removed]*: Having a minimum of 1 MeSH term guarantees that the citation extraction component can create a query of at least one MeSH term to search for in the MeSH field of Lucene's index. Thus, sentences without any MeSH terms were removed.

In the end, 404 sentences with a total of 449 citations were selected after filtering for the cases described above. Sentences with more than one citation are divided in several sentences each with an individual citation in the evaluation.

4-2 Sentence Expansion

Because our search approach in citation finding component is very word sensitive, we need to ensure that we extract all possible information based on words appearing in the input sentences. To maximize the informational match between sentence terms and the terms in relevant articles, we use Query Expansion (QE) techniques. Query Expansion is

a technique in Information Retrieval that handles the matching problem by enriching the user's query to fully reflect the information in the query. In CiteFinder, we operationalize this technique by locating important terms in the original sentence, normalizing them, and then expanding them. That is, the sentence goes through following steps:

- **Tokenization:** The OpenNLP tokenizer [46] is applied to tokenize the sentence.

- **Lexical Normalization** [47]**:** Lexical variation refers to the existence of different words or phrases expressing the same meaning, either with the same part of speech or different parts of speech. Lexical variation can involve multi-word terms (e.g. foramina magnum vs. foramen magnum), dash separated terms (e.g. inter-montane vs. intermontane), different spellings (copper sulfate vs. copper sulphate), or contain other characters (e.g. AAID's vs. AAID). The goal of lexical normalization is to replace all variations of a word with one unique form which is more likely to be found in a dictionary. Lexical normalization also involves lemmatization of the words. Lemmatization helps to match more words during the search, thereby increasing recall.

  We use the UMLS Specialist Lexicon 2013, which includes commonly occurring English words and biomedical vocabulary, including multi-word terms. The lexicon entry for each word records the syntactic, morphological, and orthographic information[1]. We obtained a list of 398,836 lexical variations from the UMLS, which also provides the normalized form for each variant. For each single and multi-word term in the input sentence, extracted as N-grams (N=1 to 5)

[1] http://www.nlm.nih.gov/pubs/factsheets/umlslex.html

from the sentence we replace the term with the corresponding normalized form in the UMLS lexicon. For example, for a sentence with 10 words, we generate 40 phrases for normalization, including 10 words as single words, 9 phrases as bi-grams, 8 phrases as tri-grams, 7 phrases as four-grams, and 6 phrases as five-grams, and replaced each of the 40 phrases with their corresponding normal form in the UMLS lexicon. The process did not remove any stop words, and was insensitive to case.

We also replaced the words with the lexical variation before indexing the articles with Lucene. This will allow the search algorithm to find the word and phrases in the database.

- **MeSH concept extraction:** MeSH is the National Library of Medicine's comprehensive controlled vocabulary thesaurus introduced in 1963. MeSH 2013 consists of 27,149 descriptors [48] in a hierarchical structure that enables searching at different levels of specificity. It also contains over 218,000 entry terms that are synonyms, alternate forms, and other closely related terms in a given MeSH record. We use entry terms to find the most appropriate MeSH heading. For example, " Heart Decompensation", “Congestive Heart Failure“ and “Congestive Heart Failure“ are some of entry terms for " Congestive Heart Failure".

  Using UMLS, we extracted 27,149 Mesh terms and 191,874 entry terms. We extracted word N-grams (n=1 to 5) from the sentence, without removing stop words and ignoring case sensitivity, and searched for the N-grams in the list of

MeSH terms as well as the list of entry term MeSH terms. If entry terms were found, the corresponding MeSH terms were added to the tokens in the sentence to expand the sentence. We only considered MeSH terms that were not in the stop words list and were at least 3 characters in length.

Apart from the citation extraction component, the added MeSH terms are also used for citation ranking (Section 4-4) and for snippet generation (Section 4-5).

- **Abbreviation expansion:** Abbreviations are used very frequently in clinical text (e.g. "CHF" as the abbreviation for "Congestive Heart Failure"), and as such, their identity of form with the corresponding expanded term is critical for the search process, in particular for our approach where the search is primarily word-based. We expanded the sentence by adding the full terms of abbreviations. Two abbreviation dictionaries relevant to cardiology were used for this purpose:

- 140 cardiology abbreviations from Cardiology Articles weblog[1]

- 121 cardiology abbreviations from NIH[2]

A total of 228 abbreviations and their full terms were obtained after removing duplicates, and these were used to expand the sentence with full terms for any occurring abbreviations. We also added abbreviations of phrases if the expanded phrase is found in the sentence. For example, if a sentence contains "Congestive Heart Failure", we expand the sentence by adding CHF to the end of

[1] http://cardiology-articles.blogspot.com/2011/01/abbreviations-in-cardiology.html
[2] http://www.ncbi.nlm.nih.gov/books/NBK2205/

the sentence. To find the expanded version of the abbreviations in the sentence, we look for 1 to 5 grams of words.

4-3 Citation Extraction

After sentences are expanded using the process described above, the next step is to find relevant citations for the (expanded) sentences, based on the MeSH terms present in the sentence. To build a fast system, we indexed MEDLINE articles with Apache Lucene version 3.0.2 [49]. Lucene is an Information Retrieval software library that provides the ability for indexing and fast search on texts.

CiteFinder stores the text, title, publication type, and MeSH terms of each article in the Lucene index format. Retrieval of articles for a sentence is done entirely using the MeSH terms occurring in the sentence. In particular, articles with at least one MeSH term in common with the sentence's MeSH terms are retrieved during this step.

4-4 Citation Ranking

In order to rank the retrieved citations with regard to their importance and similarity with the input sentence, four measures are applied: *text search*, *MeSH search*, *journal prioritization*, and *study design recognition*. We use these measures to calculate an article relevancy score for each article, then sort the set of retrieved articles using the relevancy scores, and finally, select up to 3 of the top scored articles to present to the user. In the following, we describe each of these measures and explain how we calculate the relevancy score for articles.

4-4-1 Measure 1: Text Search

The text search measure uses the words in the article to measure the similarity between the sentence and the article. We use the score calculated by Lucene for searching with the expanded sentence as the query. Lucene takes into account both the Boolean Model (BM) and the Vector Space Model (VSM) [50]. Documents approved by BM are scored by VSM with three metrics: Term Frequency (TF), Inverse Document Frequency (IDF), and Number of MeSH terms in an article.

TF is a measure which shows how many times a word appears in the document. The goal of TF is to overcome the drawbacks of the traditional Boolean model which doesn't consider term weights in queries. In comparison to the Boolean query with which the result set is often either too small or too big, using TF usually gives us more range in the ranked results and improves the results by adding more value to the major words of the document.

Traditional IR systems suffer from weighting all the words equally and thus giving more value to unimportant and invaluable terms such as stop words. In fact, some words have little or no discerning value in the text to indicate relevance to the query. IDF, or Inverse Document Frequency, on the other hand, assigns different weights to words based on how rare they are. In particular, words which occur in more documents appear to be less informative as compared to the words that appear in fewer documents (we consider them as keywords).

The traditional cosine similarity score on query q and document d is calculated by following formula:

$$Similarity(q,d) = Cosine(\vec{q},\vec{d}) = \frac{\vec{q}.\vec{d}}{|\overrightarrow{q}|.|\overrightarrow{d}|} = \frac{\sum_{i=1}^{|q|} q_i d_i}{\sqrt{\sum_{i=1}^{|q|} q_i^2}\sqrt{\sum_{i=1}^{|q|} d_i^2}}$$

* q_i and d_i are the *tf-idf* weight of term i in the query q and document d, respectively.

Using the above formula, the similarity between a sentence and an article is obtained by normalizing the length of the document by using unit vectors. This means that articles of different sizes are scored solely by word similarity, not by the length. Although for some documents, ignoring the length of the articles will not alter the results (for example, if a document consists of parts with similar content), the drawback to this approach is that larger documents with a lot of words in them are always more likely to rank better.. The reason is that longer articles contain more words and therefore have a greater chance of being matched with the words in the query. To overcome this problem, a document length normalization factor is used in Lucene which normalizes the query-document vector to a vector equal to or larger than the unit vector.

Lucene uses the following formula to rank the documents and fields based on similarity:

$$Similarity(q,d) = \frac{\vec{q}.\vec{d}}{|\overrightarrow{q}|}.\ doc-len-norm(d) = \frac{\sum_{i=1}^{|q|} q_i d_i}{\sqrt{\sum_{i=1}^{|q|} q_i^2}}.\ doc-len-norm(d)$$

4-4-2 Measure 2: MeSH Search

The MeSH search measure shows the semantic similarity of the sentence and articles. We create a query of MeSH terms by adding all of the extracted MeSH terms in one string

and use the same approach as for text search above to calculate a score by Lucene for this measure. We have discussed MeSH extraction technique in section 4-2.

4-4-3 Measure 3: Journal Prioritization

The idea behind this measure is that within a specific domain, a citation published in a high-quality journal should have a greater chance of obtaining a higher rank than a citation with the same score published in a low-quality journal. We have previously studied the task of prioritizing cardiology journals and used the obtained formula in that study in our system here [51]. In this study, all Medline abstracts retrieved by the "Congestive Heart Failure[MeSH Major Topic]" query on PubMed were downloaded. Then, using a set of rules [52] to process the XML structured abstracts, 3,443 unique email addresses of authors from US organizations were extracted. From this set, 142 cardiologists were deemed as qualified (namely, with at least 6 years of experience), and also available and willing to participate in our survey.

60 of the top cardiology journals were selected by querying PubMed for "Congestive Heart Failure[MeSH Major Topic]" and used to create a list of journals with article count. Cardiologists were asked to rate at least 20 of 60 selected journals on a scale of 1 to 5 (1=least value, 5=highest value) according to their information worth with regard to CHF. Table 3 shows the top 10 ranked journals by these cardiologists.

Table 3: Top 10 journals selected by 142 cardiologists (rating range is 1-5)

Journal names	Rating Average	Response Count
The New England journal of medicine	4.35	132
Circulation	4.35	127
Journal of the American College of Cardiology (JACC)	4.13	127
JAMA : the journal of the American Medical Association	3.86	124
Circulation. Heart failure	3.79	124
Lancet	3.74	126
JACC. Heart Failure	3.52	104
European heart journal	3.21	112
Annals of internal medicine	3.14	118
Journal of cardiac failure	3.04	117

12 journal-related numeric metrics are considered to create a Multiple Linear Regression model and find the best coefficients for a formula to rank the journals.

1. *Impact Factor (IF):* This is the number of times that published articles in the past two years in a specific journal have been cited by other substantive articles and reviews published in the same period of time, over the total number of citable articles published by the same journal in the same timeline [53].

2. *H-Index:* An article with an *h*-index of *h* has *h* articles that have at least *h* citations each [54]. The journal *h*-index can be calculated by sorting all the articles of a specific journal from a given year according to the number of times each article is cited, and then finding the highest number that is still lower than the corresponding ”number of times cited” value [55].

3. *SJR:* The SCImago Journal Ranking (SJR) uses an algorithm similar to Google's PageRank [56] to measure the average prestige of each paper of a journal [57] to indicate the scientific influence of each journal.

4. *Total Docs:* This is the total number of all types of documents published in article's journal in a specific year.

5. *Total Refs:* This is the total number of references included in the journal's published articles.

6. *3yr Docs:* This is the total number of all types of documents published in the article's journal in the past three years.

7. *3yr Cites:* This is the total number of citations in a specific year, received by the journal's documents published in the past three years.

8. *3yr Citable:* This is the total number of the journal's citable articles in the past three years.

9. *Ref/Doc:* This is the average amount of references per document for a specific year.

10. *CHF count:* This is the number of articles in MEDLINE retrieved by the "(Congestive Heart Failure[MeSH Major Topic]) AND "<journal name>"[Journal]" query for each journal.

11. *BJH*: The National Library of Medicine has provided an XML file containing information about journals[1]. The Broad Journal Heading (BJH) values for each article's journal is extracted from the journal information. If at least one of the headings contain "cardiology", we assign a value of 1, and otherwise 0.

12. *AIM*: The Abridged Index Medicus (AIM or "Core Clinical")[2] contains a list of medical science journals. We assign a value of 1 if an article's journal is in this list, and a value 0 if the journal is absent from the list.

Using the Multiple Linear Regression algorithm and trying all the possible coefficients to equalize the calculated value to the corresponding existing value obtained by the cardiologists' survey, a formula was obtained to rank each journal:

Journal Priority score =

0.82640 * SCImago Journal Ranking

– 0.00377 * Number of articles

+ 0.00258 * Number of articles for 3 years

– 0.00190 * Number of cited-articles for 3 years

– 0.01846 * Number of references per article

+ 0.00295 * Number of CHF-indexed Medline abstracts

+ 0.62864 * Is Broad Journal Heading cardiology?

– 0.32753 * Is Core clinical journal?

[1] ftp://ftp.nlm.nih.gov/online/journals/
[2] https://www.nlm.nih.gov/bsd/aim.html

Table 4 shows the top 4 journals rated by the cardiologists in the survey and a sample value of each metric used in the obtained formula (SCImago 2011 is used).

Table 4: top 4 most informative journals and value of selected metrics in the obtained formula

Journal Name	The New England journal of medicine	Circulation	Journal of the American College of Cardiology (JACC)	JAMA : the journal of the American Medical Association
SJR	9.74	5.76	7.31	4.839
Total Docs	1,808	1,094	941	1,236
3yr Docs	5,445	3,198	2,756	3,790
3yr Citable	1,844	2,199	1,559	1,177
Ref/Doc	9.52	13.01	22.9	10.64
CHF count	576	1,982	1,737	325
BJH	0	1	1	0
AIM	1	1	1	1

Our study on 2 UpToDate articles retrieved by the "Heart Failure" query showed that 94.69% of the citations in high quality articles come from the top 15 ranked journals by our obtained formula. Therefore, using this formula to take into account journal priority in the citation ranking task is helpful.

It is worth mentioning that although the formula above is specifically obtained for cardiology journals, we have shown that it is generalizable to other topics. In a similar experiment for "multiple sclerosis" (MS) as part of the same study, we reported coverage of 94.2% of citations by the top 15 ranked MS journals.

4-4-4 Measure 4: Study Design recognition

In order to determine the most appropriate evidence for the information content of a sentence, it is helpful to understand the basic design of the research studies. The study

design refers to the overall strategy that a researcher chooses to integrate the different components of the study in a coherent and logical way, thereby ensuring that the research problem is effectively addressed. It is well known that the strength of the findings in clinical research depends on the study design, following the following order: *systematic review*, *randomized controlled trial*, *multiple time series*, *nonrandomized trial*, *cohort*, *case-control*, *time series*, *cross-sectional*, and *case series* [5]. Figure 4 demonstrates the weight levels, ranging from 9 to 1, assigned to each study design type, respectively, to show the importance of the study strategy in the paper.

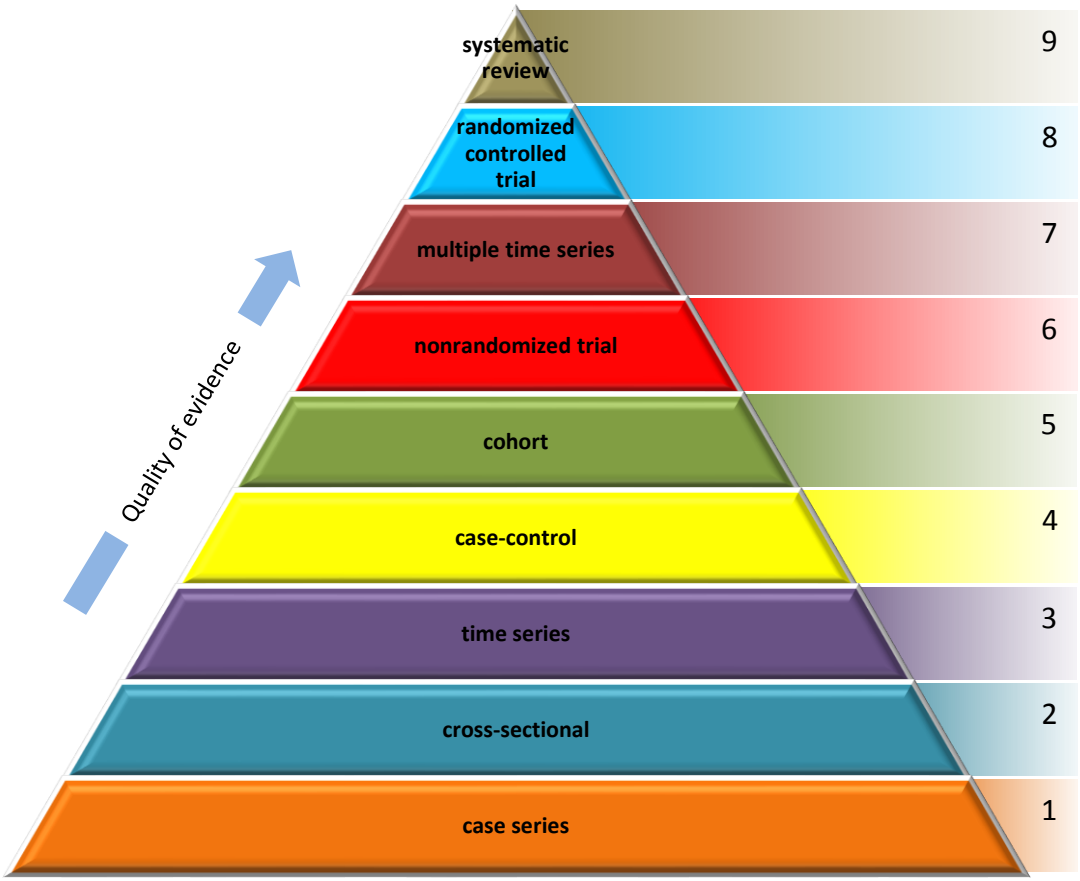


Figure 4: Study design pyramid and importance of study types

To determine the study type of a citation, we search for each study type in several sections of the articles, including publication type, abstract text, MeSH headings, and article title. These sections are selected for search because the authors usually mention the study type in these sections. Once the study type is identified, a score is assigned using the weighting shown in Figure 4. We assign a study design score of zero if none of the study type names are found the in article's sections.

Ranking Methods and Results

We propose two ranking schemes using the measures above to assign ranks to retrieved citations. It should be noted that all scores of the measures are normalized to lie within the range of 0 to 1. Score normalization is done by considering the minimum and maximum score value of each measure using the formula below:

$$Normalized\ score = \frac{(score - Min\ score)}{(Max\ score - Min\ score)}$$

To evaluate ranking methods, we consider the median rank of expected citations for each sentence in our gold standard. For example, if there are five sentences in the gold standard and the system ranked the corresponding article for each sentence as 56th, 3rd, 1st, 4th, and 33rd, the median rank of the system will be 4.

If the expected citation of a sentence is not retrieved, its rank is assumed to be the worst (lowest). So we consider the median rank of all citations in the gold standard, regardless of whether the system finds and ranks them or not. In this scenario, we were unable to find 15.81% (71 of 449) of the citations, but the currently reported median ranking is affected by recall.

1. ***Multi-Level Ranking***. A multi-level approach ranks the articles in a cascade trend. The idea is to rank the articles with one measure, and then split the sorted articles into brackets and re-rank the brackets with scores obtained from other measures. The more a measure can distinguish between articles and affect the ranking results, the sooner we use it in this system.

In the first level, the system uses the MeSH measure to extract the citations. Then, the text search measure is used to rank the articles in the first level. The reason for choosing the text search measure to rank the citations in the first (and most beneficial) level of this system is that unlike journal prioritization and study design recognition measures that are sentence independent and rank the citations regardless of the input sentence, the text search measure gives all the citations the chance to be ranked high. In contrast, if Journal prioritization or study design recognition was used in the first level, the citations would be ranked exactly the same for every sentence. This means that the first level, and therefore the whole ranking method, would prevent articles from being ranked within a wide range.

After extracting and ranking citations via the text search measure, the citations are split into N brackets based on their text search score. Note that N is a variable that can vary between "1" (all the citations are allocated to one bracket, and accordingly, the next ranking algorithm will not be applicable) and "# of retrieved citations from last level" (each citation is allocated to one bracket and therefore the ranking method to be applied afterwards has no effect). Table 5 shows the results with variations of N. Finding the best bracket size for each level is one of the challenges of this approach.

In the next step, the MeSH search measure is used to rank the citations within each bracket. Journal prioritization measure is used in the next step to rank the articles in each bracket that we obtained from MeSH search scoring.

In the last step, the study design recognition measure is used to rank the citations in each newly created N bracket to produce the final list of ranked citations.

Table 5 shows the median rank for the multi-level ranking approach.

Table 5: Multi-level ranking results

# of Brackets / Measures	10	20	100
Text search	11	11	11
Text search and MeSH search	12	11	11
Text search and journal prioritization	11	10	11
Text search and study design recognition	10	9	11
Text search, MeSH, journal prioritization, and study design recognition	11	11	11

The results – as we expected – show that none of the three measures following the text search measure add significantly to the performance of the text search measure.

2. ***Weight Ranking***. In the second approach, the final score is calculated using the following formula:

*Score = (Text search weight * Text search score) + (MeSH search weight * MeSH search score) + (Journal prioritization weight * Journal Prioritization score) + (Study Design recognition weight * Study Design recognition score)*

The idea behind the formula is to find the measures that statistically contribute to improving the results, and then find the best coefficients (weights) for each measure's score that maximizes the results (i.e., minimizes the median rank in this system).

As the first step, we obtained the median rank score for each measure individually. Table 6 shows that as expected, the text search measure contributes most to the ranking. As such, we use this measure as foundation and give it a coefficient of 1 in the formula.

Table 6: Median rank for each measure individually

Measure	Median rank
Text search	11
MeSH search	202
Journal prioritization	300
Study design recognition	243

Next, we combined other measures with the text search measure to determine which one to use in the formula.

In the first experiment, we attempted to find the best coefficient for MeSH search when the text search coefficient is 1. A range of coefficients between 0 and 2 were explored, and the results indicated 0.5 as the best weight for MeSH search. Table 7 illustrates these results.

Table 7: MeSH search coefficient impact on text search ranking (text search=1)

MeSH search Coefficient	0	0.2	0.3	0.4	0.45	**0.5**	0.55	0.6	1
Median Rank	11	10	11	10	10	**10**	10	11	15

We ran the same experiment to find the most contributing coefficient for journal prioritization on text search ranking. Table 8 demonstrates the results we achieved, with 0.35 as the best coefficient for the journal prioritization measure.

Table 8: Journal prioritization coefficient impact on text search ranking (text search=1)

Journal prioritization coefficient	0	0.2	0.3	**0.35**	0.4	0.5	1
Median Rank	11	9	8	**7**	8	8	15

Finally, we found 0.75 as the best coefficient for the study design recognition score with the same procedure as above. Table 9 shows the results.

Table 9: Study design recognition coefficient impact on text search ranking (text search=1)

Study design recognition coefficient	0	0.3	0.5	0.6	0.7	**0.75**	0.8	0.9	1	1.2	1.5
Median Rank	11	8	8	8	7	**7**	7	8	8	10	12

In order to select the measures that show statistically significant improvement on the text search ranking, we calculated the *p*-value for each of the results we obtained from the best contributing coefficient of each measure. P-value is a measure of statistical significant, obtained with *p* equal to or less than 0.01. To calculate the *p*-value for the difference in the results of a pair of measures, we used a bootstrapping method to draw 404 sentences randomly with replacement from the +citations set. We ran the text 500 times and stored the median ranks for each pair of measures. Table 10 shows the *p*-values calculated on the results of 500 runs.

Table 10: *p* value of combination of best coefficient of text search measure with other measures

measures	P value
Text search = 1 and MeSH search = 0.5	= 0.09852
Text search = 1 and journal prioritization = 0.35	< **2.2e-16**
Text search = 1 and study design recognition = 0.75	< **2.2e-16**

The results show that although the contribution of MeSH search is not significant for ranking, the improvements seen with journal prioritization and study design recognition are statistically significant. Therefore, we disregard the MeSH search measure from our formula and used the best coefficient of study design recognition (0.75) as constant to obtain the best coefficient for journal prioritization measure. Table 11 shows that the best coefficient for journal prioritization to improve the ranking is 0.45.

Table 11: Journal prioritization coefficient impact on text search and study design recognition ranking (text search=1, study design recognition=0.75)

Journal prioritization coefficient	0	0.2	0.4	**0.45**	0.5	0.6	0.7	0.9
Median Rank	7	6	4	**4**	4	5	6	8

Thus, after studying all the four measures and finding the best coefficients for the measures, we obtain the following formula to rank the articles:

$$\text{Score} = (1 * \text{text search score}) + (0.45 * \text{Journal Prioritization score}) + (0.75 * \text{Study Design recognition score})$$

4-5 Snippet Generation

Snippets are small pieces of content from retrieved documents that enable users to quickly observe the similarity of their query to each retrieved document. Snippets in

CiteFinder consist of one to three non-adjacent sequences of words from the extracted article that have the strongest contribution in the process of document retrieval for the given sentence. We also add the title and the URL of the articles to the snippet text.

A query made by a disjunction of words in the expanded sentence is given as input to Lucene and a search is executed only on the text of extracted articles to find the best fragments of text using proximity algorithms. In this step, we also eliminate articles for which the system was not able to extract any snippets. Although removing such articles can lead to a decrease in recall, it does guarantee that users will be presented with results in a consistent way, in particular with regard to the inclusion of explanatory content alongside the articles.

For snippet generation, we also explored whether using the full text for extracting snippets is better than using the abstract. The experiment on the gold standard indicated that when CiteFinder uses the full-text, it is able to extract at least one high quality snippet (two or three fragments of text) for 99.73% of citations (377 out of 378 relevant extracted and ranked citations). On the other hand, when the system looks for snippets in an abstract, it extracts high quality snippets for only 24.20% of the citations. Furthermore, for the 929020 articles that CiteFinder retrieves during the citation extraction task (Section 4-3) for all 404 sentences in +citations set, we found that the system could not generate any snippet from the abstracts for 24.91% (231488) of the articles, whereas this is true for only 0.02% (212) when the full text is used. Therefore, our study indicates that using the full text over abstracts in the snippet generation task increases the recall in both extracting high quality snippets for correctly retrieved citations, as well as in extracting at least one snippet for any retrieved citation. Although

we cannot compare the quality of snippet generation between abstracts and full text, we expect higher quality snippet generation from the full text.

4-6 Sentence Selection and Citation Assignment

In order to evaluate the two main components of the system together (sentence selection and citation assignment), we created a new gold standard system. To consider the independence paradigm in machine learning based approaches, we removed those sentences that were used for training during sentence selection, from the selected gold standard sets in the citation assignment component. 270 of 404 sentences with citations and 12,112 of 14,344 sentences without citations were thus retained in the new independent gold standard sets.

Then, we applied the sentence selection system on the candidate sentences in each set. 257 of 270 and 6,525 of 12,112 sentences in +citations and –citations sets, respectively, have been identified as sentences that need citations.

To evaluate the overall system, we ran the citation assigner system on the newly obtained +citations set and got the median rank of 12. This shows that 4.81% (13 out of 270) of the +citations sentences that were misidentified by the sentence selection system, did not affect the results significantly in the end. In other words, applying sentence selection on sentences that actually need citations, removes some of the sentences that the citation assignment component is not able to find very good citations for anyway.

We cannot run the system on the –citations set because, obviously, we do not have citations for the sentences in this set to evaluate the end results. But applying sentence

selection on this set shows that it successfully removed about half of the sentences from the primary –citations set.

4-7 Discussion

In this section, we analyze the citation assignment component and its subcomponents. We discuss areas where the system does not perform very well and ideas for improvement.

Multi-level approach

We implemented both multi-level and weight ranking algorithms to rank the citations. Results show more improvement in the weight-ranking algorithm because of the flexibility of this approach to change the effectiveness of the measures. On the other hand, the multi-level approach is sensitive to the number of results retrieved by CiteFinder. In cases where the number of retrieved articles is not considerably larger than the number of brackets, the system will not actually utilize the second- or third-level measures.

MeSH Accessibility

We use MeSH terms in the citation extraction component to extract the articles with at least one MeSH term in common with the sentence. All the articles that we have in our corpus are extracted from PubMed or PubMed Central, which provide MeSH terms for the articles. CiteFinder's limitation is that if we want to expand the corpus to cover more articles from mentioned sources, we will need to use a MeSH extractor program to pull out and index the MeSH terms from the articles. Although we are currently extracting MeSH terms from sentences in our system by using string matching methods, this method is not applicable to extract the MeSH terms of articles. More advanced methods are

needed to identify the MeSH terms that represent the whole article, not every sentence individually.

Journal Priority Measure

We studied 23 sentences related to heart failure with 31 citations. The study shows that 31% of retrieved articles (12,362 of 39,839) were not from the 63 journals we already have. Having a list of important Heart Failure–related journals will automatically guarantee that many unavailable journals are not related to the query. Even though we should assign a score of zero to them, having a complete list of journals can improve the system.

Study Design

We assigned weights of 1 through 9 to different study design types. Machine Learning algorithms can be applied to assign more accurate and meaningful weights to the elements.

Proposition identification

Since citations are retrieved solely on a MeSH term-based search, the system is not sensitive to the fact that a retrieved citation may actually be contradicting the statement of the sentence. In general, the method for citation extraction should ideally determine what proposition denoted by the sentence needs to be cited, and the search should be proposition-based and not term-based. However, when a sentence is complex, containing multiple propositions, determining which of the propositions is being cited is not evident, so developing this kind of method comes with its own set of challenges. In the future, we can implement negation detection methods to at least ensure that a retrieved citation does not contradict the input sentence.

Reference Article Collection

The major source of articles collected for the citation assignment task were different heart failure related journals accessed via PubMed, from where the articles were downloaded with the "Congestive Heart Failure[MeSH Major Topic]" query. Our initial set of journals include many top ranked ones related to heart failure, but our final selection was limited by the fact that we did not have access to the full text for most of them at the University of Wisconsin-Milwaukee (UWM). In the end, we could only retrieve full text of 533 articles from two journals for this study. In future work, we will either explore resources to purchase subscription to more journals or collaborate with other universities or institutions to expand our article collection.

Corpus independency

Prior to the work described in this thesis, we had done another study of the citation assignment task on a different gold standard and article collection [58]. Here we briefly describe the previous study and compare the results with our current results, to show that the method of obtaining a formula to rank the citations is not restricted to any specific gold standard and collection of articles. We note, though, that most of the resources used in the current study, such as MeSH terms, lexicons and abbreviations, have been updated since the earlier study.

We used the same methods to obtain 4,697 MEDLINE articles as the reference collection. We also extracted 7,864 sentences referring to 11,778 citations by querying "heart failure" on the UpToDate web site. 377 sentences referring to 456 citations were finally used for evaluation after applying the filters as described in Section 4-1.

In the previous study, we did not consider using the text search measure to rank the articles. So only MeSH search, journal prioritization, and study design recognition measures are used to obtain the following formula:

$$\text{Score} = (1 * \text{MeSH search score}) + (0.5 * \text{Journal Prioritization score}) + (0.3 * \text{Study Design recognition score})$$

This formula gave us a median rank of 41 in the citation assignment task. Detailed results using combinations of measures, methods and evaluation are reported in [58]. Applying this formula to the new corpus reported in this thesis, we obtained a median rank of 54. This shows that the algorithms we implemented in this project are isolated from the data collection and applying them on a new corpus leads to appropriate and similar results.

Chapter 5:

Conclusion and future work

Finding supporting citations for clinical sentences is challenging for clinicians. We propose a system (CiteFinder), which, after splitting the input text into sentences and identifying "needing citations" sentences, expands the selected sentences, extracts relevant citations and ranks them to retrieve the best citation for a given sentence. Our system makes it easy to generate documentation for evidence-based content by adding citations and enriching clinical content such as articles, summaries and FAQs.

This study shows that the Naïve Bayes algorithm along with words themselves (without stemming or removing stop word or considering n-Grams) as feature are the best combination in the sentence classification task to find sentences that need citations. This study also demonstrates that using journal priority and study design type will improve the text search based results by about 63% (from 11 to 4). We also show that using the full text of articles instead of just the abstract text helps in extracting better snippets.

Generalization

The proposed system explores methods to find citations for sentences in the Heart Failure domain. Further experiments will be required to check the generalizability of the system in other domains.

Text challenger

As future work, we suggest development of a system that finds articles and snippets that contradict the sentences in a given text. This system helps the authors to verify their written material before publishing them.

References

[1] "Wikipedia:About," *Wikipedia, the free encyclopedia*. 21-Apr-2014.

[2] "Smarter Decisions. Better Care.," *UpToDate*. [Online]. Available: http://www.uptodate.com/home. [Accessed: 21-Apr-2014].

[3] G. S. H. Yeo and M. L. Lim, "Maternal and fetal best interests in day-to-day obstetrics," *Ann. Acad. Med. Singapore*, vol. 40, no. 1, pp. 43–49, Jan. 2011.

[4] J. Lau, *Clinical Decision Support: The Road Ahead*. Academic Press, 2011.

[5] R. H. Fletcher, S. W. Fletcher, and G. S. Fletcher, *Clinical Epidemiology: The Essentials*. Lippincott Williams & Wilkins, 2012.

[6] R. Brandow, K. Mitze, and L. F. Rau, "Automatic condensation of electronic publications by sentence selection," *Inf. Process. Manag.*, vol. 31, no. 5, pp. 675–685, Sep. 1995.

[7] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing Text Documents: Sentence Selection and Evaluation Metrics," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 1999, pp. 121–128.

[8] Y. Gong and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2001, pp. 19–25.

[9] D. McDonald and H. Chen, "Using Sentence-selection Heuristics to Rank Text Segments in TXTRACTOR," in *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, New York, NY, USA, 2002, pp. 28–35.

[10] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 4, pp. 288–297, Jun. 1990.

[11] P. D. Turney, "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2002, pp. 417–424.

[12] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2005, pp. 115–124.

[13] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004.

[14] S. Teufel and M. Moens, "Sentence Extraction and Rhetorical Classification for Flexible Abstracts," in *IN IN TELLIGENT TEXT SUMMARIZATION*, 1998, pp. 16–25.

[15] L. McKnight and P. Srinivasan, "Categorization of Sentence Types in Medical Abstracts," *AMIA. Annu. Symp. Proc.*, vol. 2003, pp. 440–444, 2003.

[16] Z. Huang, W. Chung, T.-H. Ong, and H. Chen, "A Graph-based Recommender System for Digital Library," in *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, New York, NY, USA, 2002, pp. 65–73.

[17] Y.-L. Chen, J.-J. Wei, S.-Y. Wu, and Y.-H. Hu, "A similarity-based method for retrieving documents from the SCI/SSCI database," *J. Inf. Sci.*, vol. 32, no. 5, pp. 449–464, Oct. 2006.

[18] N. Ratprasartporn, J. Po, A. Cakmak, S. Bani-Ahmad, and G. Ozsoyoglu, "Context-based Literature Digital Collection Search," *VLDB J.*, vol. 18, no. 1, pp. 277–301, Jan. 2009.

[19] K. D. Bollacker, S. Lawrence, and C. L. Giles, "Discovering relevant scientific literature on the Web," *IEEE Intell. Syst. Their Appl.*, vol. 15, no. 2, pp. 42–47, Mar. 2000.

[20] Y. Liang, Q. Li, and T. Qian, "Finding Relevant Papers Based on Citation Relations," in *Web-Age Information Management*, H. Wang, S. Li, S. Oyama, X. Hu, and T. Qian, Eds. Springer Berlin Heidelberg, 2011, pp. 403–414.
[21] L. Plaza and A. Díaz, "Retrieval of Similar Electronic Health Records Using UMLS Concept Graphs," in *Natural Language Processing and Information Systems*, C. J. Hopfe, Y. Rezgui, E. Métais, A. Preece, and H. Li, Eds. Springer Berlin Heidelberg, 2010, pp. 296–303.
[22] W. R. Hersh and D. H. Hickam, "How well do physicians use electronic information retrieval systems? A framework for investigation and systematic review," *JAMA J. Am. Med. Assoc.*, vol. 280, no. 15, pp. 1347–1352, Oct. 1998.
[23] Z. Lu, "PubMed and beyond: a survey of web tools for searching biomedical literature," *Database J. Biol. Databases Curation*, vol. 2011, Jan. 2011.
[24] L. M. Bachmann, R. Coray, P. Estermann, and G. Ter Riet, "Identifying diagnostic studies in MEDLINE: reducing the number needed to read," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 9, no. 6, pp. 653–658, Dec. 2002.
[25] E. V. Bernstam, J. R. Herskovic, Y. Aphinyanaphongs, C. F. Aliferis, M. G. Sriram, and W. R. Hersh, "Using citation data to improve retrieval from MEDLINE," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 13, no. 1, pp. 96–105, Feb. 2006.
[26] Y. Lin, W. Li, K. Chen, and Y. Liu, "A Document Clustering and Ranking System for Exploring MEDLINE Citations," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 14, no. 5, pp. 651–661, 2007.
[27] S. J. Darmoni, L. F. Soualmia, C. Letord, M.-C. Jaulent, N. Griffon, B. Thirion, and A. Neveol, "Improving information retrieval using Medical Subject Headings Concepts: a test case on rare and chronic diseases," *J. Med. Libr. Assoc. JMLA*, vol. 100, no. 3, pp. 176–183, Jul. 2012.
[28] Z. Lu, W. Kim, and W. J. Wilbur, "Evaluation of Query Expansion Using MeSH in PubMed," *Inf. Retr.*, vol. 12, no. 1, pp. 69–80, 2009.
[29] S. Teufel, A. Siddharthan, and D. Tidhar, "Automatic Classification of Citation Function," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA, 2006, pp. 103–110.
[30] A. Siddharthan and S. Teufel, "Whose idea was this, and why does it matter? attributing scientific work to citations," in *In Proceedings of NAACL/HLT-07*, 2007.
[31] S. R. Jonnalagadda, G. Del Fiol, R. Medlin, C. Weir, M. Fiszman, J. Mostafa, and H. Liu, "Automatically extracting sentences from Medline citations to support clinicians' information needs," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 20, no. 5, pp. 995–1000, Oct. 2013.
[32] C. A. Sneiderman, D. Demner-Fushman, M. Fiszman, N. C. Ide, and T. C. Rindflesch, "Knowledge-based methods to help clinicians find answers in MEDLINE," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 14, no. 6, pp. 772–780, Dec. 2007.
[33] D.-C. Li, H. Liu, C. G. Chute, and S. R. Jonnalagadda, "Towards assigning references using semantic, journal and citation relevance," in *2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2013, pp. 499–503.
[34] Q. Li and Y. P. Chen, "Personalized text snippet extraction using statistical language models," *Pattern Recognit.*, vol. 43, no. 1, pp. 378–386, Jan. 2010.
[35] R. Xiao, Q. Hao, C. Wang, R. Cai, and L. Zhang, "Snippet Extraction and Ranking," US20110302162 A108-Dec-2011.
[36] A. Pearse, J. Douglass, and J. Moetteli, "Web snippets capture, storage and retrieval system and method," US7315848 B201-Jan-2008.
[37] S.-P. Cucerzan and M. R. Richardson, "Systems and methods that enable search engines to present relevant snippets," US7512601 B231-Mar-2009.

[38] Y. Tsegay, S. J. Puglisi, A. Turpin, and J. Zobel, "Document Compaction for Efficient Query Biased Snippet Generation," in *Advances in Information Retrieval*, M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, Eds. Springer Berlin Heidelberg, 2009, pp. 509–520.
[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor Newsl*, vol. 11, no. 1, pp. 10–18, Nov. 2009.
[40] M. F. Porter, "An algorithm for suffix stripping," *Program Electron. Libr. Inf. Syst.*, vol. 14, no. 3, pp. 130–137, Dec. 1980.
[41] T. C. Rindflesch and M. Fiszman, "The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text," *J. Biomed. Inform.*, vol. 36, no. 6, pp. 462–477, Dec. 2003.
[42] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2003, pp. 423–430.
[43] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.
[44] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer US, 2010, pp. 667–685.
[45] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Inf. Retr.*, vol. 1, no. 1–2, pp. 69–90, Apr. 1999.
[46] J. Baldridge and T. Morton, *OpenNLP*. 2004.
[47] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 17, no. 5, pp. 507–513, Oct. 2010.
[48] "Fact SheetMedical Subject Headings (MeSH®)." [Online]. Available: http://www.nlm.nih.gov/pubs/factsheets/mesh.html. [Accessed: 21-Apr-2014].
[49] "Apache Lucene." [Online]. Available: http://lucene.apache.org/. [Accessed: 18-Apr-2014].
[50] "Apache Lucene Similarity Class." [Online]. Available: http://lucene.apache.org/core/3_0_3/api/core/org/apache/lucene/search/Similarity.html. [Accessed: 12-Apr-2014].
[51] S. R. Jonnalagadda, S. Moosavinasab, D. Li, M. D. Abel, C. G. Chute, and H. Liu, "Prioritizing journals relevant to a topic for addressing clinicians' information needs," in *2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2013, pp. 5–6.
[52] S. Jonnalagadda and P. Topham, "NEMO: Extraction and normalization of organization names from PubMed affiliation strings," *J. Biomed. Discov. Collab.*, vol. 5, pp. 50–75, Oct. 2010.
[53] E. Garfield, "The history and meaning of the journal impact factor," *JAMA J. Am. Med. Assoc.*, vol. 295, no. 1, pp. 90–93, Jan. 2006.
[54] L. Bornmann and H.-D. Daniel, "Does the h-index for ranking of scientists really work?," *Scientometrics*, vol. 65, no. 3, pp. 391–392, Dec. 2005.
[55] L. Bornmann and H.-D. Daniel, "What do we know about the h index?," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 9, pp. 1381–1385, Jul. 2007.
[56] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*. 1999.
[57] B. González-Pereira, V. P. Guerrero-Bote, and F. Moya-Anegón, "A new approach to the metric of journals' scientific prestige: The SJR indicator," *J. Informetr.*, vol. 4, no. 3, pp. 379–391, Jul. 2010.

[58] S. Moosavinasab, M. Rastegar-Mojarad, L. Hongfang, and S. R. Jonnalagadda, “Towards Transforming Expert-based Content to Evidence-based Content,” *AMIA. Annu. Symp. Proc.*

Appendix: Running Example

Here we use two running examples to demonstrate input, output and results of the system in different steps. We selected two sentences with ranks of below 10 and above 300 to represent two scenarios that system leads to good or bad results.

Example 1

Input query:

> *The largest trial of nesiritide in acute heart failure found that it increased rates of hypotension, did not alter rates of death or rehospitalization at 30 days, and showed a borderline significant trend toward reducing dyspnea*

Expected citation for this sentence:

PMID:

> *21732835*

Title:

> *Effect of nesiritide in patients with acute decompensated heart failure.*

Journal name:

> *The New England journal of medicine*

MeSH terms:

- *Acute Disease*
- *Aged*

- *Double-Blind Method*
- *Dyspnea, drug therapy, etiology*
- *Female*
- *Heart Failure, complications, drug therapy, mortality*
- *Humans*
- *Hypotension, chemically induced*
- *Intention to Treat Analysis*
- *Kidney Diseases, etiology*
- *Male*
- *Middle Aged*
- *Natriuretic Agents, adverse effects, therapeutic use*
- *Natriuretic Peptide, Brain, adverse effects, therapeutic use*
- *Patient Readmission, statistics & numerical data*
- *Recurrence*

Publication type list:

- *Journal Article*
- *Multicenter Study*

- *Randomized Controlled Trial*
- *Research Support, Non-U.S. Gov't*

Abbreviations found:

heart failure → HF

Extracted MeSH terms:

Nesiritide → Natriuretic Peptide, Brain

Lexicons replaced:

heart failure → heart-failure

largest → large

found → find

increased → increase

rates → rate

rates → rate

days → day

showed → show

reducing → reduce

Expanded sentence:

The large trial of nesiritide in acute heart-failure find that it increase rate of hypotension, did not alter rate of death or rehospitalization at 30 day, and show a borderline significant trend toward reduce dyspnea HF natriuretic peptide, brain

Classified as "needing citations"?:

yes

Number of articles extracted:

3685

First ranked article only with text search measure (text=1, Mesh=0, Journal=0, Study=0):

PMID:

18039381

Title:

Outcomes of patients hospitalized for acute decompensated heart failure: does nesiritide make a difference?

Journal name:

BMC cardiovascular disorders

MeSH terms:

- *Adult*
- *Aged*

- *Aged, 80 and over*
- *Drug Costs*
- *Female*
- *Heart Failure, drug therapy, mortality*
- *Hospital Mortality*
- *Humans*
- *Length of Stay*
- *Male*
- *Middle Aged*
- *Natriuretic Agents, economics, therapeutic use*
- *Natriuretic Peptide, Brain, economics, therapeutic use*
- *Odds Ratio*
- *Retrospective Studies*
- *Treatment Outcome*

Publication type list:

- *Journal Article*
- *Multicenter Study*

Rank of corresponding article using the multi-level ranking method (order: text, MeSH, Journal, Study design):

6

Rank using the text search ranking method (text=1, Mesh=0, Journal=0, Study=0):

6

Rank using the obtained formula (text=1, Mesh=0, Journal=0.45, Study=0.75):

1

The extracted snippet from full text:

the Acute Study of Clinical Effectiveness of Nesiritide in decompensate Heart Failure (ASCEND- HF) trial ...). There were no significant difference in rate of death from any cause at 30 day (3.6% with nesiritide vs...-reported dyspnea at 6 and 24 hours, rehospitalization for heart failure or death from any cause at 30

The extracted snippet from abstract:

Nesiritide is approve in the unite state for early relief of dyspnea in patient with acute heart failure . Previous meta-analysis have raise question regard renal toxicity and the mortality associate with this agent

Example 2

Input query:

Among all patients with HF, as many as half have a normal or near normal LVEF

Expected citation for this sentence:

PMID:

12517230

Title:

Burden of systolic and diastolic ventricular dysfunction in the community: appreciating the scope of the heart failure epidemic.

Journal name:

JAMA : the journal of the American Medical Association

MeSH terms:

- *Aged*
- *Cause of Death*
- *Cross-Sectional Studies*
- *Diastole*
- *Echocardiography, Doppler*
- *Heart Failure, diagnosis, epidemiology, physiopathology*
- *Humans*
- *Middle Aged*

- *Prevalence*
- *Proportional Hazards Models*
- *Survival Analysis*
- *Systole*
- *Ventricular Dysfunction, epidemiology, physiopathology, ultrasonography*

Publication type list:

- *Journal Article*
- *Research Support, Non-U.S. Gov't*
- *Research Support, U.S. Gov't, P.H.S.*

Abbreviations found:

heart failure → HF

Extracted MeSH terms:

patients

Lexicons replaced:

patients → patient

Expanded sentence:

Among all patient with HF, as many as half have a normal or near normal LVEF heart failure patients

Classified as "needing citations"?:

yes

Number of articles extracted:

3683

First ranked article only with text search measure (text=1, Mesh=0, Journal=0, Study=0):

PMID:

19277003

Title:

DEFEAT - Heart Failure: a guide to management of geriatric heart failure by generalist physicians.

Journal name:

Minerva medica

MeSH terms:

- *Adrenergic beta-Antagonists, therapeutic use*
- *Aged*
- *Aged, 80 and over*

- *Angiotensin-Converting Enzyme Inhibitors, therapeutic use*
- *Body Fluids, physiology*
- *Cardiac Output, Low*
- *Digoxin, therapeutic use*
- *Diuretics, therapeutic use*
- *Echocardiography*
- *Family Practice*
- *Female*
- *Heart Failure, diagnosis, etiology, physiopathology, therapy*
- *Humans*
- *Male*
- *Stroke Volume, physiology*
- *Vasodilator Agents, therapeutic use*

Publication type list:

- *Case Reports*
- *Journal Article*
- *Research Support, N.I.H., Extramural*

Rank of corresponding article using the multi-level ranking method (order: text, MeSH, Journal, Study design):

355

Rank using the text search ranking method (text=1, Mesh=0, Journal=0, Study=0):

563

Rank using the obtained formula (text=1, Mesh=0, Journal=0.45, Study=0.75):

335

The extracted snippet from full text:

Context Approximately half of patient with overt congestive heart failure (CHF) have diastolic... a major criterion if it occur in response to therapy for congestive heart failure (CHF). A patient ... among those with moderate or severe diastolic or systolic dysfunction, les than half had recognize

The extracted snippet from abstract:

Approximately half of patient with overt congestive heart failure (CHF) have diastolic dysfunction without reduce ejection fraction (EF). Yet, the prevalence of diastolic dysfunction and its relation to systolic dysfunction and CHF in the community remain undefined