Dakota State University
Beadle Scholar

Masters Theses & Doctoral Dissertations

Spring 3-1-2016

Advanced Data Analytics for Systematic Review Creation and Update

Prem Timsina
Dakota State University

ADVANCED DATA ANALYTICS FOR SYSTEMATIC REVIEW CREATION AND UPDATE

A dissertation submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Doctor of Science

in

Information Systems

March 2016

By
Prem Timsina

Dissertation Committee:
Dr. Jun Liu (Co-chair)
Dr. Omar El-Gayar (Co-chair)
Dr. Nevine Nawar
Dr. Viki Johnson

DAKOTA STATE UNIVERSITY

DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Science in Information Systems degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or the university.

Student Name: Prem Timsina

Dissertation Title: Advanced Data Analytics For Systematic Review Creation And Update

Dissertation Chair: Date: 4/27/2016

Dissertation Chair: Date: 4/27/2016

Committee member: Viki Johnson Date: 4/28/2016

Committee member: Nevine Nawar Date: 4 | 29 | 2016

ACKNOWLEDGMENT

Optimism is the faith that leads to achievement. Nothing can be done without hope and confidence. –**Helen Keller**

ABSTRACT

Evidence Based Medicine (EBM) refers to the application of state-of-the-art medical evidence to improve the quality and reduce the cost of medical care. While systematic reviews (SRs) are positioned as an essential element of modern evidence-based medical practice, the creation and update of these reviews is a much more demanding, rigorous, and resource-intensive process than developing a literature review in other domains. Specifically, systematic reviews attempt to bring a high level of rigor to reviewing research evidence and are often developed based on a peer-reviewed protocol so that they can be replicated if necessary.

To support the update of existing systematic reviews, we investigate various supervised learning techniques, feature extraction techniques, and sampling techniques to resolve class imbalance issue. Specifically, we used soft-margin Support Vector Machine (SVM) as a classifier, exploited Unified Medical Language Systems (UMLS) for medical terms extraction, and examined various techniques to resolve the class imbalance issue. Through an empirical study, we demonstrate that soft-margin SVM achieves better classification performance than the existing algorithms used in current research, and the performance of the classifier can be further improved by using UMLS to identify medical terms in articles and applying re-sampling methods to resolve the class imbalance issue.

For supporting the creation systematic reviews, we explore semi-supervised learning based classifiers to identify articles that can be included when creating medical systematic reviews (SRs). Specifically, we perform comparative study of various semi-supervised learning algorithm, and identify the best technique that is suited for SRs creation. We also aim to identify whether semi- supervised learning technique with few labeled samples produce meaningful work saving for SRs creation. The results indicate that semi-supervised learning could significantly

reduce the human effort and is a viable technique for automating medical systematic review creation with a small-sized training dataset. We also demonstrate the viability of semi-supervised learning algorithm along with self-learning and active learning when training dataset is rare, which is often the practical case in many machine-learning problems.

From a theoretical perspective, this research explores the possibility of machine-learning techniques in new domain (Systematic Review generation), particularly as it relate to the creation of systematic reviews, the use of semi-supervised learning, and the use of full-text in the creation and update of systematic reviews. The experiences and lessons learned from this research are expected to inform the literature regarding the efficacy of the proposed techniques and the further development and refinement of these techniques. From a practical and applied research perspective, this research is expected to result in a significant reduction in the cost of creating and updating systematic reviews. Overall, this research improves the availability of best medical evidence, and consequently, can positively and significantly impact the health and wellbeing of society. This research can be extended to other areas as well such as education, ecommerce, business, finance, etc.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Prem Timsina

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

1.1. Background

Evidence Based Medicine (EBM) refers to the application of state-of-the-art medical evidence to improve the quality and reduce the cost of medical care (A. Cohen et al. 2010). Although the classical vision of EBM required physicians to directly search the relevant medical research for evidence applicable to their patients, the modern conception of EBM heavily relies on synthesis of research findings in the form of an evidence report commonly referred to as a systematic review (SR). According to Higgins and Green (Higgins and Green 2011), "a systematic review is a high-level overview of primary research on a particular research question that tries to identify, select, synthesize and appraise all high quality research evidence relevant to that question in order to answer it". Each systematic review addresses a clearly formulated problem. As an example, (Couch et al. 2008) presents a systematic review of "diabetes education for children with Type 1 Diabetes Mellitus and their families". It synthesizes the findings presented in 80 pertinent articles. Nowadays, systematic reviews form a key resource for informing evidence based medical practice. With the increasingly rapid pace by which medical knowledge is created, researchers, practitioners and policy makers are challenged to keep pace with state-of-the-art medical evidence and incorporate such evidence into practice. Systematic reviews respond to this issue by recognizing, appraising, and synthesizing research-based evidence from multiple sources and presenting it in an accessible format (Mulrow 1994).

Developing a medical systematic review is a much more demanding, rigorous, and resource-intensive process than developing a literature review in other domains, since systematic reviews attempt to bring a high level of rigor to reviewing research evidence and are often developed based on a peer-reviewed protocol so that they can be replicated if necessary. Surprisingly, the current workflow for creating and updating SRs is largely a manual process. An initial search by querying databases such as Medline, Cochrane and Embase often returns a large number of articles given a medical topic. Developing the review presented in (Couch et al. 2008) first involves retrieving 12,740 articles based on keywords such as diabetes, diabetic children, diabetic family members, and diabetes education in order to ensure that none of the relevant articles will be missed. These 12,740 articles were then evaluated manually by a team of scientists using highly methodic procedures. Only 80 of them were selected according to the inclusion and exclusion guidelines. Finally, the scientists synthesized the research findings in the 80 articles to establish the best education for children with Type 1 diabetes mellitus and their families. The articles that need to be included in a systematic review are usually selected in two steps. The first step is called abstract triage, where scientists identify "relevant" articles that can potentially be included in a SR based on the title and abstract of the articles. This phase of screening articles usually requires a long time and significant effort as it involves a group of scientists evaluating thousands of articles in order to find the relevant ones. The second step is referred to as full-text triage. It involves full text inspection of the relevant articles selected in the title/abstract triage to determine those that satisfy the inclusion criteria and will be included in a systematic review (Shojania et al. 2007). Due to the manual workflow of selecting articles for systematic reviews

(SRs), developing SRs requires a significant investment in time (1,139 expert hours on average) and funds (up to a quarter of a million dollars) from a dedicated and qualified research team (McGowan and Sampson 2005a).

Nowadays, medical knowledge base is growing at an astounding pace. Reports of new clinical trials are being published at the rate of over 20,000 per year (Shojania et al. 2007). This creates an enormous challenge for scientists trying to develop and update systematic reviews to keep pace with the development in the medical field. Cochrane Collaboration estimates that at least 10,000 new systematic reviews are needed to cover most of the healthcare problems (Higgins and Green 2011). Unfortunately, fewer than half of this number has been published even after ten years of focused effort by the EBM community (Higgins and Green 2011). Once a review is created, the job is not done yet - a systematic review needs to be updated periodically (Higgins and Green 2011). The median time for a review to become obsolete is 5.7 years; for some medical conditions like cardiovascular, a SR may be obsolete in less than a year (Shojania et al. 2007). A report published by Agency for Healthcare Research Quality (AHRQ) indicates that only 2% of systematic reviews published in all journals represent updates of previous reviews (whether conducted by the same authors or not) (Shojania et al. 2007). Researchers have attributed the difficulty of developing and updating systematic reviews to keep up with medical research advances to the aforementioned resource intensive manual process needed to screen articles (Shemilt et al. 2013). We lack highly refined automated tools that help reviewers sort and prioritize articles, which has become a bottleneck that has hitherto constrained the timely creation and update of systematic reviews.

There are efforts that have leveraged text analytics (Adeva et al. 2014; Shemilt et al. 2013) to automate the article screening procedure for systematic reviews. Most existing literature focuses on addressing a text classification problem, where medical articles are classified as relevant or irrelevant to the topic based on the title and abstract of the articles. As in any text classification task, we need to enhance both recall (i.e., among the articles that are deemed relevant and included in a systematic review, the fraction of those classified as "relevant") and precision (i.e., among the articles that are classified as "relevant", the fraction of those will actually be included in a review). Any automated system for identifying relevant articles must maintain a very high level of recall since a systematic review should include most, if not all, articles that provide high quality evidence relevant to the topic. Any system with a low recall would be of little use (Matwin et al. 2010). Precision is also essential in this context since a higher precision means that the articles that are classified as relevant are indeed relevant, which means that a smaller number of articles needed to be reviewed during the downstream full-text triage stage. Hence, in order to resolve the aforementioned bottleneck in the screening of medical articles, it is necessary to improve precision while maintaining a high recall. Among the existing research, a few studies such as (Bekhuis and Demner-Fushman 2012; A. M. Cohen et al. 2009; Matwin et al. 2010) attempted to achieve a high recall. Nonetheless, the results of these studies have shown a tendency for precision to decline as recall increases. Another conspicuous issue that has been largely ignored in existing research is that systematic review datasets are normally highly imbalanced, which means that among the thousands of articles to be selected, only a small number of them will be included in the final systematic review. The imbalance ratio ranges from 1:10 to

1:1,000(Shemilt et al. 2013). Class imbalances have been reported to hinder the performance of classifiers proposed in existing research (Bekhuis and Demner-Fushman 2012; A. M. Cohen et al. 2009; Matwin et al. 2010). Another issue that has been highly diregarded by existing research is the unavailability of training dataset for creation of new medical systematic review. Most of the existing research use supervised learning assuming readily available training data and focus on updating reviews. For example, Cohen et al. (2006) used 50% training and 50% validation data, Adeva et al. (2014) used 90% training and 10% validation data, and other studies have embraced a similar approach. However, supervised machine learning assumes the availability of training data sets that do not necessarily exist when creating SR. A need exist to explore other approaches that are more suited to situations where training data sets are not readily available, e.g., when creating SR.

1.2. Objective of the Research

The objective of this research is to develop computer methods for automatically classifying articles for inclusion and exclusion for systematic review creation and update. We hypothesize that by exploiting powerful advanced analytics techniques and a suitably constructed, high quality small-set of training data, we can construct a classifier to automate the article triage procedure for SR creation and update. For creating systematic reviews, we investigate semi-supervised learning method, in which a small set of labeled data is used for training machine-learning algorithm. Next, we investigate ensemble techniques and class imbalance issues to optimize accuracy and create a generalizable model. For updating systematic reviews, we investigate a supervised-learning method that uses the original SR as its training dataset. When successfully completed, this research has the potential to facilitate the creation of over 5,000 new SRs, which are immediately needed to cover new medical problems and allows for the update of 98% of existing reviews. From a theoretical perspective, this research contributed to the machine learning and text analytics literature by exploring, adapting, and developing approaches to a new problem domain, namely the classification and update of medical literature for the purpose of creating and updating systematic reviews.

1.3. Problem Identification

Issues Specific to Systematic Review Update: The median time for an SR to become obsolete is 5.7 years; nevertheless, for some conditions like cardiovascular, the SR obsolete time is less than a year (Shojania. et al. 2007). Therefore, keep up with recent developments, a review should undergo periodic update. Despite identifying a need for periodic SR updates, the scenario of SR updates in literature is not satisfactory. According to an Agency for Healthcare Research Quality (AHRQ) report, two-thirds of survey participants reported that over 20% of the reviews they created are obsolete, and the remaining one-third of participants reported that 50% of reviews they commissioned are obsolete (Shojania. et al. 2007). Overall, only 2% of systematic reviews published in all journals represent updates of previous reviews (whether conducted by the same authors or not) (Shojania. et al. 2007). In brief, despite the need and importance of evidence updating, it is apparent that current practice for updating these reviews has not kept pace (Moher et al. 2007). One of the main reasons is attributed to resource limitations (Shojania. et al. 2007). The problem is exacerbated by the by the need for more frequent updates. According to AHRQ, update guidelines such as updating every two years may miss significant contributions that occur within shorter time lines. Accordingly, future research should aim to devise automated methods to continuously monitor newly published articles related to the original systematic review. Based on the quantity and quality of new articles, researchers can determine the need for a systematic review update (Shojania. et al. 2007).

Issues Specific to Systematic Review Creation: Cochrane Collaboration estimates that at least 10,000 new SRs are needed to cover most of the healthcare problems (A. Cohen et al. 2010). Unfortunately, fewer than half this number has been published even after ten years of focused effort by the EBM community (A. Cohen et al. 2006). This is commonly attributed to resource intensive manual process need to create and update these reviews (McGowan and Sampson 2005a). In response to this situation, there have been some attempts in the literature to reduce the workload of manual screening. However, the need for a substantial amount of labeled dataset for the training purpose is the most critical problem in literature for SR creation. To create a labeled data set, experts need to manually review a considerable number of articles to produce a training set For example, Frunza et al. used 20,000 articles as the training set and 27,274 articles as the test set requiring the manual review of 20,000 articles just to train the machine learning algorithm.

Issues Common to Systematic Review Creation and Update: One of the crucial issue in this area is satisfactory recall and accuracy. The best and most consistent outcome of existing research was recall of 94.6% and accuracy of 17% (Frunza et al. 2010), i.e., if 20,000 articles are retrieved in certain SRs, then, 16,600 articles must be still manually reviewed. This, means existing research has not substantially optimized the workload, which is a crucial step for machine learning being viable technique for systematic review generation.

1.4. Research Questions

The over-arching research questions related to the research problems discussed above are:

1- *Issues Specific to Systematic Review Creation:* How can we create a machine-learning model for SR creation with a minimum or no training dataset?

2- *Issues Specific to Systematic Review Update:* How can we quantify the characteristics of included articles for a particular SR topic? How can we use the characteristics to identify newly published articles? How can we triage articles for an SR update?

3- *Issues Common to SR Creation and Update:* How can we create a generalized machine learning model? How can we boost the accuracy and recall of article classification models?

We aim to answers research questions by performing the comparative investigation of various supervised and semi-supervised based learning approach, and investigating techniques to resolve class imbalance issues and exploring the possibility of using Unified Medical Language System (UMLS) for text classification. The details of these research questions are explained in chapter 2 (systematic review update) and chapter 3 (systematic review creation) of this dissertation.

The dissertation is organized as follows: the Chapter 2 discusses the proposed approach for automating the update of systematic reviews. Specifically, we illustrate the motivation, dataset, methodology and empirical results regarding the systematic review update. In the following chapter we focus on the process of creating systematic reviews. While in the last chapter, we present our concluding remarks of application of machine learning in systematic review update and creation and highlight the theoretical and practical contribution of this thesis report.

2. Systematic Review Update

2.1. Introduction

Nowadays, medical knowledge base is growing at an astounding pace. Reports of new clinical trials are being published at the rate of over 20,000 per year (Shojania et al.). This creates an enormous challenge for scientists trying to develop and update systematic reviews to keep pace with the development in the medical field. Cochrane Collaboration estimates that at least 10,000 new systematic reviews are needed to cover most of the healthcare problems (Higgins and Green). Unfortunately, fewer than half of this number has been published even after ten years of focused effort by the EBM community (Higgins and Green). Once a review is created, the job is not done yet - a systematic review needs to be updated periodically (Higgins and Green). The median time for a review to become obsolete is 5.7 years; for some medical conditions like cardiovascular, a SR may be obsolete in less than a year (Shojania et al.). A report published by Agency for Healthcare Research Quality (AHRQ) indicates that only 2% of systematic reviews published in all journals represent updates of previous reviews (whether conducted by the same authors or not) (Shojania et al.). Researchers have attributed the difficulty of developing and updating systematic reviews to keep up with medical research advances to the aforementioned resource intensive manual process needed to screen articles (Shemilt et al.). We lack highly refined automated tools that help reviewers sort and prioritize articles, which has become a bottleneck that has been hitherto constrained the timely creation and update of systematic reviews.

There are efforts that have leveraged text analytics (Adeva et al. 2014; Bekhuis and Demner-Fushman 2012; Shemilt et al. 2013) to automate the article screening procedure for systematic reviews. Most existing literature focuses on addressing a text classification problem, where medical articles are classified as relevant or irrelevant to the topic based on the title and abstract of the articles. As in any text classification task, we need to enhance both recall (i.e., among the articles that are deemed relevant and included in a systematic review, the fraction of those classified as "relevant") and precision (i.e., among the articles that are classified as "relevant", the fraction of those will actually be included in a review). Any automated system for identifying relevant articles must maintain a very high level of recall since a systematic review should include most, if not all, articles that provide high quality evidence relevant to the topic. Any system with a low recall would be of little use (Matwin et al.). Precision is also essential in this context since a higher precision means that the articles that are classified as relevant are indeed relevant, which means that a smaller number of articles needed to be reviewed during the downstream full-text triage stage. Hence, in order to resolve the aforementioned bottleneck in the screening of medical articles, it is necessary to improve precision while maintaining a high recall. Among the existing research, a few studies such as (Bekhuis and Demner-Fushman 2012; A. Cohen et al. 2006; Matwin et al. 2010) attempted to achieve a high recall. Nonetheless, the results of these studies have shown a tendency for precision to decline as recall increases. Another conspicuous issue that has been largely ignored in existing research is that systematic review datasets are normally highly imbalanced, which means that among the thousands of articles to be selected, only a small number of them will be included in the final systematic review. The imbalance ratio ranges from 1:10 to 1:1,000 (Shemilt et al. 2013). Class imbalances have been reported to hinder the performance of classifiers proposed in existing research. (Bekhuis and Demner-Fushman 2012; A. Cohen et al. 2006; Matwin et al. 2010)

The objective of this chapter is to develop an advanced analytics-based approach to automatically identifying relevant articles that could be included in systematic reviews based on the title and abstract of the articles while updating exiting medical systematic review report. Our text analytics based approach aims to improve the precision of article classification for systematic reviews while sustaining a very high level of recall. It makes three improvements to the existing methods described in literature. First, we propose to use the Unified Medical Language Systems (UMLS) to extract medical terms as features for article classification, while the majority of existing research uses the "bag-of-words" approach (Adeva et al. 2014; Bekhuis and Demner-Fushman 2012; A. Cohen et al. 2006; Shemilt et al. 2013; Wallace et al. 2010) Our study demonstrated that the automatically extracted Unified Medical Language System (UMLS) terms helped boost classification performance. Second, we propose to use soft-margin polynomial Support Vector Machine (SVM) to classify articles. Using different medical datasets, we showed that soft-margin polynomial SVM achieved higher precision and recall, compared with several algorithms proposed in existing research. Third, to deal with the aforementioned class imbalance problem, we examined various re-sampling methods to re-sample the training data. The results of our comparative experiments indicate that a soft-margin polynomial SVM classifier that leverages more precise feature representation using UMLS and integrates the Synthetic Minority Oversampling (SMOTE) method (Chawla

2010) has the potential to yield significantly improved performance in identifying relevant articles for systematic reviews.

2.1.1. Related work

There have been some attempts in literature to leverage analytics to automate systematic reviewer update (Ananiadou et al. 2009; Bekhuis and Demner-Fushman 2012; A. Cohen et al. 2006; Frunza et al. 2010; Shemilt et al. 2013). One of the most significant research done in this area is one conducted by Cohen et al. (2006). In this National Institute of Health (NIH) supported project, Cohen et al. used the perceptron algorithm to identify journal articles for inclusion in systematic reviews based on the title and abstract of the articles. While the perceptron-based classifier achieved high recall, precision was consistently low. By fixing recall to be at least 95%, it produced very low precisions when applied to a number of datasets such as Antihistamines (precision = 0%), SkeletalMuscleRelaxants (precision = 0%), and Triptans (precision = 3.65%).

Adeva et al.'s research (2014) is probably the most comprehensive one so far in this area. They conducted experiments that involved multiple classification algorithms (including naïve Bayes, KNN, Support vector machines, and Rocchio) combined with several feature selection methods (including TF, DF, IDF, etc.) and applied to different parts of the articles (including the titles alone, abstracts alone and both titles and abstracts). SVM has been proved to produce the best performance with respect to the F1 scores. Bekhuis and Demner-Fushman (2012) also compared different algorithms including K-nearest neighbor (KNN), naïve Bayes, complement naïve Bayes (cNB), and evolutionary SVM (EvoSVM) (implemented in the RapidMiner) and used information gain as their feature selection method to select features from article titles and abstract.

EvoSVM has been proved to be the most effective among the algorithm. One reason SVM and its variations often outperform other algorithms is that a medical document is normally represented as a feature vector with words or phrases as the features for classification. This feature vector is often high dimensional and sparse; that is, for each document, its feature vector only has a few entries that are non-zero. SVM has the potential to handle large number of features with overfitting protection (Joachims 1998a), and it works well with problems with sparse features (Kivinen et al. 1995). Similar to Cohen et al. (2006), Bekhuis and Demner-Fushman's study (2012) also proved the inverse relationship between precision and recall. Precision was maximal when recall was very low, e.g., precision=100% and recall=7.69%. When maintaining a high recall (100% for two datasets, ameloblastoma and influenza), evoSVM, though the best among the tested algorithms, produced relatively low precisions (13.11% for the ameloblastoma dataset and 10.69% for the influenza dataset).

As mentioned previously, class imbalance remains a critical, yet largely ignored issue in this context. (Shemilt et al. 2013) is perhaps the only research that investigated the use of re-sampling in selecting articles for systematic reviews. They used undersampling by drawing a random sample of excluded records equal in number to the total number of records marked as provisionally eligible for inclusion and proved that undersampling helps enhance that the performance of the text-mining based classifiers (Shemilt et al. 2013) In addition to undersampling, oversampling techniques, though never used in the area of systematic reviews, have long been proved to be effective in dealing with class imbalance in data mining literature. For instance, Ling et al. (1998) combined oversampling of the minority class with undersampling of the majority class and concluded that the best results are obtained when both classes are equally represented. A particular type of oversampling, namely the Synthetic Minority Oversampling Technique (SMOTE) (Chawla 2010), creates synthetic examples of the minority class instead of just randomly duplicating minority examples. Chawal et al. (2010) conducted various experiments with different datasets and proved that SMOTE outperforms plain undersampling and oversampling, and furthermore, the combination of SMOTE and undersampling performs even better than SMOTE alone. It is hence intriguing to investigate if re-sampling techniques such as SMOTE can help improve the performance of article classification in the context of systematic reviews.

Overall, the findings of extant research show enough promise to further consider the possibility of using data analytics techniques for automatically screening articles for systematic reviews (A. Cohen et al. 2006; Frunza et al. 2010; Shemilt et al. 2013; Tsafnat et al. 2014). However, further research is needed to develop appropriate classifiers, resolve the class imbalance problem, and improve the precision of classification techniques while maintaining a high recall.

2.1.2. Research Gap

Our literature review indicates that 1) for any automated classification technique to be of practical use in supporting article selection for systematic reviews, it is critical for the technique to achieve a high level of recall, and 2) it is necessary to improve precision while sustaining a high recall since a higher precision means that fewer articles would need to be manually reviewed in the downstream full-text triage stage. Improving precision while sustaining a high recall, however, is a difficult task, as shown in existing research. This leads us to the following overarching research question:

How can we develop a classification technique that helps improve precision while sustaining a high recall (above 95%)?

We plan to address this research question by investigating which combination of textual analytics techniques is most valuable in identifying relevant articles that should be included in a systematic review.

Existing research into automatic article classification for systematic reviews has almost exclusively relied on the bag-of-words approach for feature representation. While this de facto standard has led to promising results, we feel that other feature extraction schemes may provide better predictive ability. Prior research (Aronson et al. 2007; H. Liu et al. 2002), though not in the area of systematic reviews, has corroborated the observation that biomedical text classification can be improved by enriching raw text with automatically extracted Unified Medical Language System (UMLS) terms. As an example, Kilicoglu et al. (2009) demonstrated the feasibility of automatically identifying "scientifically rigorous" articles using multiple features from publications, including "high-level" features such as Unified Medical Language System (UMLS) terms. This leads us to the following research question:

Can we improve precision while sustaining a high recall by using automatically extracted Unified Medical Language System (UMLS) terms as features?

As discussed previously, the issue of class imbalance is critical, yet not sufficiently addressed in this field. To address the issue, Cohen et al. (2006) modified the conventional perceptron algorithm by adjusting the false-negative learning rate (FNLR) to improve the recall to be over 95%. Another possible approach is using re-sampling methods to re-sample the training data. In the area of data mining, various re-sampling strategies such as undersampling, oversampling and SMOTE oversampling, have been proposed to classify datasets with highly asymmetric positive and negative sample frequency. It is hence meaningful to investigate:

Can we use a re-sampling method to further improve precision while sustaining a high recall?

2.2. Methodology

Our analytics approach to identifying relevant articles for systematic reviews includes three major components: 1) feature extraction using the UMLS, 2) soft-margin polynomial SVM, and 3) SMOTE combined with undersampling. We conducted experiments using four systematic review datasets and compared analytics techniques with others that were proposed in existing research. In following sub-sections, we describe the data sources, each component in our analytics approach, and the methods that we compared our techniques with in detail.

2.2.1. Data Sources

We used four systematic reviews on drug topics including ACEInhibitors (ACE), Antihistamines (AN), Skeletal-MusleRelaxants (SKE), and Triptans (TRIP), performed by AHRQ's Evidence-based Practice Center (EPC) at Oregon Health and Science University as our datasets. These four systematic review datasets were also used in (A. Cohen et al. 2006). Cohen et al. (2006) defined a new measure WSS@95%, i.e., percentage of work saved when recall is fixed to be at least 95%, to measure the effectiveness of the perceptron-based classifier. The perceptron-based classifier proposed in (A. Cohen et al. 2006) turned very low WSS@95% values (0.00%, 0.00% and 3.37) and low precisions (3.87%, 0.00%, and 3.65%) on three of the four dataset AN, SKE and

TRIP, respectively, when maintaining the recalls to be over 95%. We hence used these datasets in our experiments since we intended to investigate if our proposed approach can help improve the precision and WSS@95% values. The perceptron-based classifier achieved relatively high performance (recall =95.61%; WSS@95% = 56.61%) but low precision (3.87%) for the dataset ACE. We included this dataset in our study to investigate if our approach helps achieve comparable or better WSS@95% by enhancing precision. The original datasets include the PubMed Unique Identifiers (PMID) of all the articles and the inclusion and exclusion decisions made by human researchers. Following (A. Cohen et al. 2006), we focus on classifying the articles based on the title and abstract of the articles. We used Medline's Batch Entrez features to extract the title and abstract of all the articles based on their PMIDs. Table 1 shows an overview of the datasets. As discussed above, imbalanced class distributions are the norm for article selection in systematic reviews. Only a small ratio of articles has been included in each of the four systematic reviews. Among the four dataset, SkeletalMuscleRelaxants has the most serious class imbalance problem with only 9 included articles. Consequently, the perceptron-based algorithm proves to be ineffective with precision = 0.55% (classify everything in one class) and WSS@95% recall (defined later) = 0.00% for the dataset.

Table 1: Overview of Data Corpus—Systematic Review Update

Dataset	Total number of articles	Number of excluded articles	Number of included articles	Ratio—Included vs. Excluded
ACEInhibitors (ACE)	2544	2503	41	1:61
Antihistamines (AN)	310	294	16	1:18
SkeletalMuscleRelaxants (SKE)	1643	1634	9	1:182
Triptans (TRIP)	671	647	24	1:26

2.2.2. Feature extraction

We used the MEDLINE records for each article in the four datasets to generate the feature set as input to our classification technique. The feature set includes the features extracted from the title and abstract as well as the article's Medical Subject Headings (MeSH) and MEDLINE publication type. To extract features from the title and abstract of an article, we propose to use the UMLS to automatically extract terms and use them as features. Most of the existing research has relied on the "bag-of-words" approach to extracting features. We conducted experiments to compare the performance between these two methods for feature extraction (i.e., UMLS vs bag-of-words). Below we briefly describe both methods.

The features extracted from the bag-of-words approach used in our comparative experiments included not only unigrams (i.e., individual words) but also 2-term and 3-term n-grams. Each document (i.e., a text file including the article tile and abstract) is represented by a vector of weights m features:

$$d_j = (w_{1j}, w_{2j}, \dots \dots, w_{mj})$$

where m is the number of features, and w_i is the weight of the i^{th} features (including unigrams, 2-grams and 3-grams). The weight value of a feature represents how much that feature contributes to the semantics of the document $d_{j\,.}$ If there are n documents in total, the corpus is represented by n*m matrix, which is usually called term-document matrix. In a term-document matrix, if a certain feature (i.e., a word) does not occur in the document, then the weight of that feature becomes 0 for that document. Following (Bekhuis and Demner-Fushman 2012), we used the method TF-IDF(term frequency / inverse document frequency) (Robertson 2004)to create the weights. TF-IDF is a numerical statistic that reveals the importance of a feature in a document in a dataset. The TF-IDF value of a word increases as it appears more often in a document; however, the TF-IDF value is offset by the frequency of the word in the whole dataset. This helps to mitigate for the fact that some words such as “patient” are generally more common than other words in medical documents.

We propose to extract features from the titles and abstracts using the UMLS Metathesaurus. UMLS allows to extract terms from different vocabularies, including CPT, ICD-10-CM, LOINC, MeSH, RxNorm, and SNOMED CT. Moreover, UMLS enables us to extract the Concept Unique Identifier (CUIs), semantic types, and synonymous terms used in medical literature (US National Library of Medicine 2014), We used the MetaMap program that maps words and phrases to different UMLS semantic types. An example of UMLS terms extracted from an abstract is given below.

The free medical text appears as:

> “The objective of this study was to examine the relationships of serum and dietary magnesium (Mg) with prevalent cardiovascular disease (CVD), hypertension, diabetes mellitus, fasting insulin, and average carotid intimal-medial wall thickness measured by B-mode ultrasound.”

The UMLS terms and their semantic types appear as:

Study Objective [Idea or Concept]
Relationships [Qualitative Concept]
Serum (Specimen Source Codes - Serum) [Intellectual Product]
Serum (Specimen Type - Serum) [Body Substance]
Dietary Magnesium [Element, Ion, or Isotope]
Cardiovascular (Cardiovascular system) [Body System]
disease prevalence (disorder prevalence) [Quantitative Concept]
Hypertension (Hypertension Adverse Event) [Finding]
Diabetes Mellitus [Disease or Syndrome]
fasting (Act Code - fasting) [Intellectual Product]
Insulin [Amino Acid, Peptide, or Protein,Hormone,Pharmacologic Substance]
Insulin (Recombinant Insulin) [Amino Acid, Peptide, or Protein,Hormone,Pharmacologic Substance]
Average [Quantitative Concept]
Carotid [Body Part, Organ, or Organ Component]
Intima [Tissue]
Medial [Spatial Concept]
Wall (Walls of a building) [Manufactured Object]
Thickness (Thick) [Qualitative Concept]

Measured [Qualitative Concept]
ultrasound b mode (B mode ultrasound) [Diagnostic Procedure]
MEASURED (Measured Tumor Identification) [Diagnostic Procedure]
ultrasound b mode (B mode ultrasound) [Diagnostic Procedure]

We used the UMLS-extracted terms as the features for our classifier. For instance, in the example shown above, the terms such as “Study Objective”, “Serum (Specimen Source Codes - Serum)” “Cardiovascular (Cardiovascular system)”, “fasting (Act Code - fasting)”, etc. have been used as features for classification. In our experiments, we compared the UMLS-based feature extraction method with the conventional bag-of-words approach described above.

2.2.3. Algorithms

We propose to use soft-margin polynomial SVM to enhance the classification performance and compare it with other algorithms that have proved to be effective in existing research. In order to explain soft-margin polynomial SVM, we describe the regular “hard-margin” SVM algorithm first.

SVM with liner kernel: Existing studies such as (Bekhuis and Demner-Fushman 2012; Joachims 1998b; H. Liu et al. 2002)has proved the effectiveness of SVM with a linear kernel in text classification in the process of medical systematic reviews. The optimization problem associated with SVM is shown below.

$$min_{\mathbf{w},b}\frac{\mathbf{w}^{\mathbf{T}}\mathbf{w}}{2}$$

$$\text{subject to: } y_i(\mathbf{w}^T\mathbf{x_i}+b)\geq 1 \ (\forall \text{ data points } \mathbf{x_i}).$$

where for each data point (x_i, y_i), y_i is either 1 or −1, indicating the class to which the point belongs. The two hyperplanes w · x – b = 1 and w · x – b = -1 are called support vectors that separate the data. SVM maximizes the distance (called “margin”) between the support vectors.

***Soft-margin polynomial SVM*:** We propose to use the soft-margin Support Vector Machine (SVM) with a polynomial kernel as a classifier. Soft-margin polynomial SVM is an extension of the standard “hard” margin SVM described above.

The “hard-margin” SVM sometimes does not work well since it does not allow data points in the margin. However, data is not often perfectly linearly separable, and it is necessary to allow some data points of one class to appear within the region bounded by the support vectors. Soft-margin polynomial SVM provides the flexibility by introducing a slack variable $\epsilon_i \geq 0$, and the optimization problem of soft-margin polynomial SVM becomes (Stanford 2014):

$$min_{\mathbf{w},b,\epsilon} \frac{\mathbf{w^T w}}{2} + C \sum_i \epsilon_i$$
$$\text{subject to: } y_i(\mathbf{w}^T \mathbf{x_i} + b) \geq 1 - \epsilon_i \text{ and } \epsilon_i \geq 0 \ (\forall \text{ data points } \mathbf{x_i}).$$

where ϵ_i, the slack variable, represents the degree of error in classification. The optimization hence becomes a tradeoff between a large margin and a small error penalty (i.e., ϵ_i). When the training set is not linearly separable, and there exists no hyperplane that can perfectly separate positive and negative samples, the optimization results in a “soft” margin that may contain some misclassified data points. The parameter C known as a regularization term can be seen as a method for controlling overfitting - it is tradeoff between the importance of maximizing the margin and fitting the training data. That is, if the C value is large, then model is better fitted to the training data (may cause over-fitting), whereas if the C value is small, SVM fits on the bulk of data (Cortes and Vapnik 1995). In our experiments, when applying soft-margin SVM to each dataset, we selected the best performing C and ϵ value that help maximize precision while sustaining recall to be over 95%, based on cross-validation.

***evoSVM*:** Bekhuis and Demner-Fushmanb (Bekhuis and Demner-Fushman) found that evoSVM achieved best performance, compared with KNN, naïve Bayes, complement naïve Bayes (cNB**)** (Bekhuis and Demner-Fushman). evoSVM is a SVM implementation using an evolutionary algorithm (ES) to solve the dual optimization problem of a SVM. In our experiments, following Bekhuis and Demner-Fushmanb (2012), we used the Rapid-Miner's implementation of evoSVM and followed the evoSVM settings recommended by the authors: radial kernel; Gaussian mutation; gamma= 1.0; epsilon = 0.1; and C = 1.

***Perceptron**:* Cohen et al. (2006) used a perceptron-based classifier to predict when articles should be added to existing drug class systematic reviews. A perceptron is a type of neural network that finds a linear function to discriminate between classes. In essence, a single layer perceptron is simply a linear classifier, which is efficiently trained by a simple rule: It starts with an initial set of guessed weights (i.e. numerical parameters), and then for all wrongly classified data points, the weights either increase or decrease to reduce the prediction errors.

Naïve Bayes: Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. According to Adeva et al. (2014), naïve Bayes seemed to provide the best results in terms of false negatives. We hence also included this algorithm in our comparison.

2.2.4. Re-sampling methods

We examined four re-sampling techniques for resolving the aforementioned class imbalance issue.

Undersampling reduces the number of samples in the majority class in the training set until the ratio between the minority class and the majority class is at a desired level (H. Liu et al. 2002). Theoretically, researchers cannot control what information of a majority class is thrown away. Also, undersampling is often problematic since important information about the decision boundary between the majority and minority class may be eliminated (A. Y.-c. Liu 2004). One of the benefits of undersampling is its very simple implementation. The overall number of samples in a training set is greatly reduced, which means that training time is greatly reduced. In our research, we randomly selected a portion of the majority class, which in our case are the articles excluded from the systems reviews, so that the number of excluded articles in each sampled dataset is equal to that of the included articles. For example, in the ACEInhibitors dataset, there are 1,252 excluded articles that were excluded from and 21 included articles. We re-sampled the articles in the training dataset and created a new training set that includes all 21 included articles and 21 randomly selected excluded articles.

Oversampling seeks to increase the number of samples in the minority class by replicating samples from that class (He and M 2013). The advantage of this approach is that less information from the majority class is lost, as compared to undersampling. The primary disadvantage of this approach is that it tends to over fit the trained model. In our experiments, we tested different oversampling rates including 100% (i.e., replicating the minority samples once), 200% (i.e., replicating the minority samples twice), 300% (i.e., replicating the minority samples three times), and 400% (i.e., replicating the minority samples four times). We stopped at 400% oversampling because our experiments showed that the classifier started to suffer from over-fitting on all four datasets. We then select the best performing oversampling rate (among 100%, 200%, 300% and 400%) based on cross-validation for each dataset.

The Synthetic Minority Oversampling Technique (SMOTE) proposed in (Chawla 2010) is different from the conventional oversampling method described above. The conventional oversampling method oversamples the minority class by randomly replicating minority examples. This affects the decision region of the minority class, which results in a similar but more specific region in the feature space (Chawla 2010). In the SMOTE, the minority class is oversampled by creating synthetic examples rather than replicating the minority class examples. In our experiments, we oversampled the minority class by taking each minority class example and developing synthetic examples along the line segments joining any k minority class nearest neighbors (in our case five neighbors). For example, if the rate of oversampling is 200%, only two neighbors among the five nearest neighbors will be randomly chosen, and a synthetic sample will be generated for each neighbor. If the oversampling rate is 300%, then for each example in the training dataset, three of its neighbors will be randomly selected, and three synthetic samples will be generated. Synthetic samples are computed according to the following procedure described in (Chawla 2010): 1) compute the difference between the sample under consideration and its nearest neighbor, 2) multiply the difference by a random number between 0 and 1, and 3) add the result from 2) to the feature vector under consideration to create a synthetic sample. We tested SMOTE using different oversampling rates including 100%, 200%, 300% and 400% to oversample the minority class and selected the best oversampling rate for each dataset based on cross-validation.

A combination of SMOTE and undersampling: We considered combining both SMOTE and undersampling. We investigated combinations of different undersampling rates and SMOTE rates, including 1) 50% undersampling of the majority class + 100% SMOTE of the minority class, 2) 50% undersampling of the majority class + 200% SMOTE of the minority class, 3) 75% undersampling of the majority class + 100% SMOTE of the minority class, 4) 75% undersampling of the majority class + 200% SMOTE of the minority class, and 5) undersampling of the majority class + 200% SMOTE of the minority class to make the ratio between the majority and minority classes be 1. Again, we selected the best performing combination of sampling rates for each dataset based on cross-validation.

2.3. Evaluation Methods

We evaluated the classification performance using four metrics, precision, recall, F1-score and Work Saved over Sampling at 95% confidence interval or WSS@95% in short, a metric proposed in (A. Cohen et al. 2006). These measures are defined based on a confusion matrix as shown in Table 2. In our research, we treated the articles that were included in a review as positive examples and those that were excluded as negative examples. TP represents the number of True Positives, i.e., positive examples that were correctly classified by our SVM classifier. TN is the number of True Negatives, i.e., negative examples that were correctly classified, FP the number of False Positive, i.e., negative examples that were incorrectly classified as positive, and FN the number of False Negatives, i.e., positive examples incorrectly classified as negatives.

Table 2: Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	True negative (TN)	False positive (FP)
Actual Positive	False negative (FN)	True positive (TP)

The formulas for computing recall, precision, F1 and WSS@95% are shown in Table 3. Recall refers to the rate of correctly classified positives among all positives and is equal to TP divided by the sum of TP and FN. Precision refers to the rate of correctly classified positives among all examples classified as positive and is equal to the ratio of TP to the sum of TP and FP. F1 means the harmonic mean of recall and precision. WSS@95% is defined as percentage of examples that meet the initial search criteria and do not need to be manually reviewed because they have been correctly classified. Setting recall above 95%, WSS can be calculated as the ratio of the sum of TN and FN to the total number of samples minus 0.05.

Table 3: Evaluation metrics—Systematic Review Update

Evaluation Metric	Formula
Recall	TP/ (TP+FN)
Precision	TP/(TP+FP)
F1	(2*recall*precision)/(recall + precision)
WSS@95%	(TN + FN)/N – 0.05
N= Total Number of Samples in Positive and Negative Classes **WSS@95%= Work Saved over Sampling at 95% confidence interval**	

It is noteworthy that we do not use accuracy or AUC (area under ROC curve) as evaluation metrics for two reasons. First, when the class distribution is imbalanced, the evaluation based on accuracy breaks down. For instance, in the dataset SkeletalMuscleRelaxants, if a classifier classifies all articles (4 positive articles and 817 negative articles) as negative, then the predicted accuracy would be 99.51%. A very high accuracy rate is achieved without detecting any articles that should be included. Second, classification accuracy assumes equal misclassification costs (for false positive and false negative errors), which is problematic because one type of classification error often can be more expensive than another. In classification for systematic reviews, the cost of false negative is high because we intend to avoid missing any articles that should be included in a systematic review. According to Cohen et al. (2006), any analytics models that achieve a recall less than 95% is meaningless. Therefore, we preset the recall of a positive class to be greater than 95%, and we examined approaches to improve the precision of the algorithm. Precision defines the fraction of retrieved documents classified as relevant that are indeed relevant. The higher the precision, the smaller number of articles scientists need to manually review.

To make the most efficient use of the datasets and to get the best estimate of system performance on future data, we chose to follow (2006) and used 5×2 cross-validation. In 5×2 cross-validation, the data set is randomly split in half, and then one half is used to train the classifier, and the classifier is scored using the other half as a test set. Then the roles of the two half data sets are exchanged, with the second half used for training and the first half used for testing, with the results accumulated from both halves of the split (Dietterich 1998). What makes 5 × 2 different from the ten-way cross-validation more commonly used is that the half-and-half split and score process is repeated five times. This approach uses each data sample five times for training and five times for testing among random splits and averages the results together for all runs.

2.4. Experimental Procedures

We conducted two experiments to evaluate the effectiveness of our approach. The datasets we used in the experiments are the four datasets we described in section 2.2.1 including ACEInhibitors (ACE), Antihistamines (AN), SkeletalMuscleRelaxants (SKE) and Triptans (TRIP). The detail of our experiment design is illustrated in Table 4.

Experiment 1 consists of two steps. First, we used the unigrams, 2-grams and 3-grams extracted from article titles and abstracts using the bag-of-words approach plus the Medical Subject Headings (MeSH) and MEDLINE publication type as the features and compared soft-margin polynomial SVM with other algorithms including SVM with linear kernel, evoSVM, naïve Bayes, and perceptron. Second, we used the automatically extracted UMLS terms plus the MeSH and MEDLINE publication type as the features. Experiment 1 was designed to compare the performance of soft-margin polynomial SVM against the other algorithms. We also compared the effectiveness of the UMLS-based feature extraction against the bag-of-words method.

Table 4. Overview of experiments—Systematic Review Update

		Features	Algorithms	Sampling method
Exp. 1	Step 1	Bag-of-words (up to 3-grams) extracted from titles and abstracts + Medical Subject Headings (MeSH) + MEDLINE publication type	Soft-margin polynomial SVM, SVM, EVO-SVM, Perceptron, Naïve Bayes	N/A
	Step 2	UMLS terms extracted from titles and abstracts + Medical Subject Headings (MeSH) + MEDLINE publication type		
Exp. 2	Step 1	Bag-of-words (up to 3-grams) extracted from titles and abstracts + Medical Subject Headings (MeSH) + MEDLINE publication type	Soft-margin polynomial SVM	No sampling Undersampling, Oversampling, SMOTE, SMOTE + Undersampling
	Step 2	UMLS terms extracted from titles and abstracts + Medical Subject Headings (MeSH) + MEDLINE publication type		

After identifying soft-margin polynomial SVM as the most effective algorithm in Experiment 1, we conducted Experiment 2 to investigate if different re-sampling methods including undersampling, oversampling, SMOTE, and SMOTE combined with undersampling can further enhance the classification performance. We also conducted Experiment 2 in two steps. In Step 1, we used features extracted using the bag-of-words approach, and in Step 2, we used the UMLS extracted features. In both steps, we used soft-margin polynomial SVM as the classifier, combined with different re-sampling methods.

2.5. Experimental Results and Discussion of Findings

2.5.1. Experiment 1 results

Step 1. In this step we compared multiple algorithms with the features extracted using the bag-of-words approach plus the MeSH and MEDLINE publication type. The results of this step are shown in Table 5 with the highest performance measures for each dataset being highlighted.

As discussed previously, we intend to improve precision while sustaining a high recall. According to (A. Cohen et al. 2006), a recall of 0.95 or greater is required for an automated classification system to identify an adequate fraction of the relevant articles. However, among the five algorithms we investigated, two of them, naïve Bayes and evoSVM, do not have sufficient configuration options that allow us to fix recall to be over 95%. We fixed recall to be at least 95% for the other three algorithms including soft-margin polynomial SVM, SVM with linear kernel and perceptron. To do so, drawing upon (A. Cohen et al. 2006), we fixed the false-positive learning rate at 1.0 and adjusted the false-negative learning rate (FNLR) to optimize performance for each dataset. We tested different FNLRs in a consistent manner across for each dataset and applied cross-validation to identify the optimal FNLR that resulted in an as-high-as-possible precision while maintaining over 95% recall.

Table 5: Experiment 1 step 1 results—Systematic Review Update

Dataset	Algorithm	N	TP	TN	FP	FN	Precision	Recall	F1-sore	WSS@95%
ACE	Soft-margin SVM	1273	21	809	442	1	4.53	95.45	8.65	58.55
	SVM	1273	21	81	1171	1	1.76	95.45	3.46	1.44
	Perceptron	1273	21	775	476	1	4.23	95.45	8.10	55.87
	evoSVM	1273	14	635	617	8	2.22	63.64	4.29	0.00
	Naïve Bayes	1273	7	1245	7	15	50.00	31.82	38.89	0.00
AN	Soft-margin SVM	156	9	28	119	0	7.03	100.00	13.14	12.95
	SVM	156	9	16	131	0	6.43	100.00	12.08	5.26
	Perceptron	156	0	0	147	9	0.00	0.00	0.00	0.00
	evoSVM	156	4	42	105	5	3.67	44.44	6.78	0.00
	Naïve Bayes	156	2	142	5	7	28.57	22.22	25.00	0.00
SKE	Soft-margin SVM	809	5	191	613	0	0.81	100.00	1.61	18.61
	SVM	809	0	804	5	0	0.00	0.00	0.00	0.00
	Perceptron	809	0	0	804	5	0.00	0.00	0.00	0.00
	evoSVM	809	3	318	486	2	0.61	60.00	1.21	0.00
	Naïve Bayes	809	1	803	1	4	50.00	20.00	28.57	0.00
TRIP	Soft-margin SVM	338	13	107	218	0	5.62	100.00	10.64	26.65
	SVM	338	13	74	254	0	4.19	100.00	8.042	16.89
	Perceptron	338	13	28	297	0	4.19	95.83	8.02	3.28
	evoSVM	338	7	117	208	6	3.26	53.85	6.15	0.00
	Naïve Bayes	338	9	283	42	4	17.65	69.23	28.13	0.00

Among the three algorithms where we achieved recall of at least 0.95, including soft-margin polynomial SVM, SVM with linear kernel (shown as SVM in Table 5), and perceptron, the soft-margin polynomial SVM was prominent in achieving 100% recall for three of the four datasets (including AN, SKE and TRIP) and 95.45% for the dataset ACE. SVM with linear kernel returned high recalls in three datasets (including ACE, AN, and TRIP), but failed to identify any positive examples, thus resulting in 0% recall and precision for SKE. The Perceptron algorithm achieved high recalls (95.45.61% and 95.83%) for ACE and TRIP, but produced 0% recall and precision for the other two datasets. Among these three algorithms with fixed recall, soft-margin polynomial SVM achieved the highest precision (4.53% in ACE, 7.03% in AN, 0.81% in SKE, and 5.62% in TRIP) and the highest F1 scores for all four datasets. Soft-margin polynomial SVM also returned the highest WSS@95% (56.9% in ACE, 13% in AN, 18.6 % in SKE, and 26.65% in TRIP) for all four datasets. Perceptron returned the second highest WSS@95% (55.87%) for the dataset ACE, and SVM with linear kernel returned the second highest WSS@95% (16.83%) for TRIP. It is noteworthy that in the case of the dataset SKE, where both SVM with linear kernel and perceptron failed to identify any true positive (TP) examples (see Table 5) and hence returned 0% precision and 0% WSS@95%, soft-margin polynomial SVM was able to produce 18.61% work reduction. Also, for the dataset AN, soft-margin SVM produced 12.95% WSS@95%. It was followed by SVM with linear kernel with a much lower WSS@95% (5.26%). Among the four datasets we used, SKE and AN have a smaller number of positive examples (16 and 9 respectively). Our soft-margin SVM appeared to be more effective than the other algorithms in dealing with datasets with a small number of positive articles.

If we consider all of these five algorithms, soft-margin polynomial SVM produced the second highest F1 scores for all four datasets. On the surface, naïve Bayes appeared to have achieved higher precisions and F1 scores. For instance, naïve Bayes returned a high precision (28.57%) and the highest F1 score (25.00%) but a low recall (22.22%) when applied to the dataset AN. However, a close investigation revealed that it returned only two true positive predictions, which means among the nine articles that were included in a systematic review, the naïve Bayes classifier has classified only two of them to be positive. Similarly, for the dataset SKE, naïve Bayes achieved relatively high precision (50%) and highest F1 score (28.57%), but made only one true positive prediction. This proves that for asymmetrically distributed datasets, precisions and F-scores are not meaningful when a high recall cannot be obtained. The experimental results in Step 1 clearly showed that among the five algorithms we have compared, soft-margin polynomial SVM achieved the best performance when we used the features extracted using the bag-of-words approach. Moreover, soft-margin polynomial SVM performed significantly better than the other algorithms for the datasets that have a small number of positive examples.

Table 6: Experiment 1 step 2 results—Systematic Review Update

Dataset	Algorithm	N	TP	TN	FP	FN	Precision	Recall	F1-sore	WSS@95%
ACE	Soft-margin SVM	1273	21	1065	186	1	10.41	**5.45**	8.34	**7** **8.74**
	SVM	1273	21	500	751	1	2.72	**5.45**	.29	3 4.36
	Perceptron	1273	21	865	386	1	5.16	**5.45**	.79	6 3.03
	evoSVM	1273	13	1113	138	9	8.61	9.09	5.03	0. 00
	Naïve Bayes	1273	15	1225	26	7	36.59	8.18	**7.62**	0. 00
AN	Soft-margin SVM	156	9	24	123	0	6.82	**00.00**	2.77	**1** **0.38**
	SVM	156	9	18	129	0	6.52	**00.00**	2.24	0. 53
	Perceptron	156	0	147	0	9	0.00	.00	.00	0. 00
	evoSVM	156	2	137	10	7	16.67	2.22	9.05	0. 00
	Naïve Bayes	156	2	138	9	7	18.18	2.22	**0.00**	0. 00
SKE	Soft-margin SVM	809	5	436	368	0	1.34	**00.00**	.65	**4** **8.89**
	SVM	809	0	804	0	5	0.00	.00	.00	0. 00
	Perceptron	809	0	804	0	5	0.00	.00	.00	0. 00
	evoSVM	809	3	764	40	2	6.98	0.00	**2.50**	0. 00
	Naïve Bayes	809	2	770	34	3	5.56	0.00	.76	0. 00
TRIP	Soft-margin SVM	329	13	173	152	0	7.88	**00.00**	4.61	**4** **6.18**
	SVM	329	13	122	203	0	6.02	**00.00**	1.35	3 1.09
	Perceptron	329	13	107	218	0	5.63	**00.00**	0.66	2 6.66
	evoSVM	329	11	136	189	2	5.50	4.62	0.33	0. 00
	Naïve Bayes	329	5	309	16	8	23.81	8.46	**9.41**	0. 00

Step 2. In this step, we compared multiple algorithms with features including the automatically extracted UMLS terms plus the Medical Subject Headings (MeSH) and

MEDLINE publication type. Table 6 shows the performance of the five algorithms. Again, evoSVM and naïve Bayes returned recall values below the acceptable level (95%) for all datasets. Among the other three algorithms with over 95% recall, soft-margin polynomial SVM had the highest precision across all four datasets. It also had 100% recall for three datasets (AN, SKE and TRIP) and 95.45% recall for the ACE dataset. SVM with linear kernel produced 95.45% recall for the ACE dataset, but soft-margin polynomial SVM achieved higher precision (10.14% vs. 2.72%) and much higher WSS@95% (78.74% vs. 34.36%). Among the three algorithms with fixed recall, soft-margin polynomial SVM again produced the highest precisions and F1 scores for all of the four datasets. Soft-margin SVM distinguished itself from the other algorithms when applied to the dataset SKE that has only 9 positive examples. While all the other algorithms resulted in 0% work saved, soft-margin SVM produced 48.89% WSS. Naïve Bayes had the highest precision and F1 scores; however, the low recalls rendered the precisions and F1 scores hardly meaningful. Our findings in step 2 of experiment 1 are consistent with those obtained in step1. Soft-margin SVM performed better than the other algorithms across all four datasets when we used the automatically extract UMLS terms as the features. It was the optimal algorithm that could provide an improved precision and enhanced percentage of work saved, especially when applied to datasets with few positive examples.

Comparing the results obtained in step 1 vs. step 2, we found that when applied to three datasets including ACE, SKE and TRIP, all three algorithms with recall fixed to be at least 95% achieved higher precisions, F1 scores and WSS@95% when UMLS was used to extract features. These three algorithms, however, achieved overall worse results for the dataset AN. Table 7 shows the performance of soft-margin polynomial SVM using the UMLS terms as features vs. using bag-of-words. For the dataset AN, soft-margin SVM successfully identified all included articles in the dataset, but it performed slightly worse with a larger FP value (123 vs. 119), which is not critical given that reviewers just need to manually review 4 additional articles. Using UMLS to extract features significantly enhanced the performance of the soft-margin SVM classifier when applied to the other three datasets. A possible reason behind the UMLS-based feature extraction method outperforming the bag-of-words approach is that the bag-of-words features are created by extracting n-grams from articles without considering the semantics of the words. UMLS (used in conjunction with vocabularies such as CPT, MeSH, SNOMED CT, etc.), on the other hand, identifies the semantic type for each extracted term and provides the synonyms of the term when available. Moreover, using UMLS to extract terms entails an automatic variable selection procedure - it extracts only the terms that are commonly used in medical literature. This automatic variable selection helps improve classification performance by reducing over-fitting.

Table 7: Comparing soft-margin SVM results obtained in step 1 vs those obtained in step 2—Systematic Review Update

Dataset	Feature extraction method	N	TP	TN	FP	FN	Precision	Recall	F1-sore	WSS@95%
ACE	Bag-of-words	1273	21	789	463	1	4.34	95.45	8.30	56.90
	UMLS	1273	21	1065	186	1	10.41	**5.45**	**8.34**	**7** **8.74**
AN	Bag-of-words	156	9	28	119	0	7.03	100.00	13.14	13.06
	UMLS	156	9	24	123	0	6.82	100.00	12.77	10.38
SKE	Bag-of-words	809	5	191	613	0	0.81	100.00	1.61	18.6
	UMLS	809	5	436	368	0	1.34	100.00	2.65	48.89
TRIP	Bag-of-words	338	13	107	218	0	3.63	100.00	10.66	26.65
	UMLS	338	13	173	152	0	7.88	100.00	14.61	46.18

In summary, the results of experiment 1 demonstrated that 1) soft-margin polynomial SVM consistently performed better than the other algorithms across the four datasets, and 2) overall, using the UMLS terms as features helps enhance the performance of soft-margin polynomial SVM and the other algorithms as well.

2.5.2. Experiment 2 results

Table 8: Experiment 2 step 1 results with features extracted based on bag-of-words—Systematic Review Update

Dataset	Sampling method	N	TP	TN	FP	FN	Precision	Recall	F1-sore	WSS@95%
ACE	Undersampling	1273	21	859	392	1	5.08	95.45	9.66	62.56
	Oversampling	1273	21	853	398	1	5.01	95.45	.52	62.09
	SMOTE	1273	21	952	299	1	6.56	95.45	2.28	69.86
	SMOTE + Undersampling	1273	21	981	270	1	7.22	95.45	**3.42**	72.14
	Non-sampling	1273	21	809	442	1	4.53	95.45	.65	58.55
AN	Undersampling	156	9	4	143	0	5.92	100.00	11.18	0.00
	Oversampling	156	9	21	126	0	6.67	100.00	2.50	8.46
	SMOTE	156	9	22	125	0	6.72	100.00	12.59	9.10
	SMOTE + Undersampling	156	9	21	126	0	6.67	100.00	2.50	8.46
	Non-sampling	156	9	28	119	0	7.03	100.00	**3.14**	12.95
SKE	Undersampling	809	5	107	697	0	0.71	100.00	1.41	8.23
	Oversampling	809	5	317	487	0	1.02	100.00	.01	34.18
	SMOTE	809	5	434	370	0	1.33	100.00	2.63	48.65
	SMOTE + Undersampling	809	5	400	404	0	1.22	100.00	.42	44.44
	Non-sampling	809	5	191	613	0	0.81	100.00	.61	18.61
TRIP	Undersampling	338	13	43	282	0	4.41	100.00	8.44	7.72
	Oversampling	338	13	134	191	0	6.37	100.00	1.98	34.64
	SMOTE	338	13	164	161	0	7.47	100.00	3.90	43.52
	SMOTE + Undersampling	338	13	201	124	0	9.49	100.00	**7.33**	54.47
	Non-sampling	338	13	107	218	0	5.62	100.00	0.64	26.65

After demonstrating that soft-margin SVM is the better classification algorithm compared with the other algorithms in Experiment 1, we investigated if we can further enhance precision while maintaining a high recall using different re-sampling methods. We tested four re-sampling technique - undersampling, oversampling by replicating minority class examples, SMOTE, and SMOTE combined with undersampling. Again, we conducted the experiment in two steps. In both steps, we used soft-margin SVM as the classifier.

Step 1. In this step, we used the bag-of-words extracted features plus the Mesh and MEDLINE publication type as the features. We compared the four different sampling methods including undersampling, oversampling by replicating minority class examples, SMOTE, and SMOTE combined with undersampling. Table 8 shows the results obtained in this step. It also includes the performance measures of soft-margin SVM when no re-sampling has been conducted (shown as "non-sampling" in Table 8).

Undersampling means that we randomly select a subset of the negative examples (articled excluded from the systematic reviews in this case), so that the number of positive examples is equal to that of the positive examples. When compared with non-sampling, undersampling was only able to produce the improved performance for the dataset ACE (62.5% WSS@95). It failed to achieve improved performance for both SKE and TRIP. Undersampling did not work at all for the dataset AN. It helped to improve the classification performance for the dataset ACE due to the fact that there are relatively a large number of positive examples, which might be sufficient to train the classifier. We then oversampled the minority class examples (i.e., the included articles). For each dataset, we selected the optimal sampling rate based on the method described in section 4.4. Oversampling by replicating the minority class examples (shown as "oversampling" in Table 8) enhanced classification performance with respect to the F1 score and

WSS@95% for three datasets including ACE, SKE and TRIP. It worked especially well for the dataset SKE with only 9 positive examples. SMOTE is another oversampling technique for increasing the number of minority class examples. Compared with non-sampling, SMOTE showed significantly improved performance for two datasets SKE and TRIP. It boosted WSS@95% from 18.61% to 48.65% for SKE and from 26.65% to 43.52% for TRIP.

Table 9: Experiment 2 step 2 results—Systematic Review Update

Dataset	Sampling method	N	TP	TN	FP	FN	Precision	Recall	F1-sore	WSS@95%
ACE	Undersampling	1273	21	1065	186	1	10.82	95.45	19.44	78.74
	Oversampling	1273	21	936	315	1	6.50	95.45	12.17	68.61
	SMOTE	1273	21	1096	155	1	12.88	95.45	22.70	81.17
	SMOTE + undersampling	1273	21	1104	147	1	13.55	95.45	23.73	81.80
	Non-sampling	1273	21	960	264	1	7.72	95.45	14.29	70.49
AN	Undersampling	156	9	6	141	0	6.00	100.00	11.32	0.00
	Oversampling	156	9	22	125	0	6.72	100.00	12.59	9.10
	SMOTE	156	9	38	109	0	7.63	100.00	14.17	19.36
	SMOTE + undersampling	156	9	43	104	0	7.96	100.00	14.75	22.56
	Non-sampling	156	9	24	123	0	6.82	100.00	12.77	10.38
SKE	Undersampling	809	5	349	455	0	1.15	100.00	2.28	38.14
	Oversampling	809	5	516	342	0	1.56	100.00	3.08	58.78
	SMOTE	809	5	478	326	0	1.64	100.00	3.24	54.09
	SMOTE + undersampling	809	5	630	174	0	3.29	100.00	6.37	72.87
	Non-sampling	809	5	436	368	0	1.45	100.00	2.85	48.89
TRIP	Under-sampling	338	13	62	263	0	4.78	100.00	9.12	13.34
	Oversampling	338	13	204	121	0	10.00	100.00	18.18	55.36
	SMOTE	338	13	215	110	0	10.92	100.00	19.70	58.61
	SMOTE + under-sampling	338	13	220	105	0	11.40	100.00	20.47	60.09
	Non-sampling	338	13	173	152	0	8.07	100.00	14.94	46.18

As shown in Table 9, SMOTE also outperformed plain oversampling across all four datasets. Combining SMOTE and under-sampling enabled our classifier to achieve higher precisions, F1 scores and WSS@95% than SMOTE alone for two datasets including ACE and TRIP. It produced slightly worse performance for the other two datasets. The datasets ACE and TRIP have larger numbers of included articles than the other two datasets, which indicates that with the bag-of-words features, SMOTE combined with undersampling may be the optimal re-sampling method when applied to datasets with relatively a large number of positive examples, while we may need to use SMOTE alone when dealing with datasets with a small number of positive examples. It is also noteworthy that for the dataset AN, the classifier without any re-sampling achieved the best performance.

Step 2.In this step, we used the UMLS terms as the features. Again, we compared the four different sampling methods including undersampling, plain oversampling, SMOTE oversampling, and SMOTE combined with undersampling. Table 9 shows the results we obtained in this step.

With the UMLS terms as the features, the classifier with undersampling showed performance that is consistent with what we obtained in Step 1. It did not work at all for the dataset AN. Compared with non-sampling, undersampling failed to improve performance for three datasets except ACE. Different form the results we obtained from Step 1, for AN, both SMOTE alone and SMOTE combined with undersampling produced better precision and WSS@95% values than non-sampling. It is noteworthy that SMOTE combined with undersampling appeared to be the best re-sampling method for all four datasets. It worked particularly well for the dataset SKE with only 9 positive examples. It doubled the precision produced by SMOTE alone and raised the WSS@95% value from 54.09% to 72.87%.

In Table 10, we compared the best performing re-sampling methods obtained in Step1 and in Step 2. With the automatically extracted UMLS terms as the features in Step 2, SMOTE combined with under-sampling achieved better performance for all four datasets, and it worked particularly well for AN and SKE.

Table 10: Comparing soft-margin SVM results obtained in step 1 vs those obtained in step 2—Systematic Review Update

Dataset	Step	Best sampling method	N	TP	TN	FP	FN	Precision	Recall	F1-score	WSS@95%
ACE	1	SMOTE + Undersampling	1273	21	981	270	1	7.22	95.45	13.42	72.14
	2	SMOTE + undersampling	1273	21	1104	147	1	13.55	95.45	23.73	81.80
AN	1	Non-sampling	156	9	28	119	0	7.03	100.00	13.14	12.95
	2	SMOTE + undersampling	156	9	43	104	0	7.96	100.00	14.75	22.56
SKE	1	SMOTE	809	5	434	370	0	1.33	100.00	2.63	48.65
	2	SMOTE + undersampling	809	5	630	174	0	3.29	100.00	6.37	72.87
TRIP	1	SMOTE + Undersampling	338	13	201	124	0	9.49	100.00	17.33	54.47
	2	SMOTE + under-sampling	338	13	220	105	0	11.40	100.00	20.47	60.09

In summary, the results of experiment 2 demonstrated that 1) overall, SMOTE-based re-sampling methods including both SMOTE alone and SMOTE combined with undersampling helped improve classification performance of the soft-margin SVM classifier, whether we used the UMLS extracted features or bag-of-words; 2) the combination of SMOTE and undersampling in general performed better than SMOTE alone when the UMLS terms were used as the features. It is understandable that undersampling failed to achieve high performance since in undersampling, we make the ratio between the positive class and the negative class equal to 1 by reducing the number of negative examples, thus losing considerable amounts of information from the negative examples. SMOTE in general outperformed plain oversampling because in plain oversampling, the decision region that results from classification of the minority class actually becomes smaller as we replicate the minority class examples. SMOTE offers more related minority class examples to learn from, which leads to more coverage of the minority class, thus allowing a learner to create broader decision regions (Chawla 2010). Moreover, oversampling tends to cause overfitting because of repetitive instances that tightens the decision boundary. In contrast, with artificially created examples, SMOTE softens the boundary region and is hence less susceptible to overfitting (Longadge et al. 2013).

Finally, following suggested data mining practice (T. Y. Liu et al. 2007) we compared our analytics techniques with an existing benchmark model. The benchmark we used is the perceptron model developed in Cohen et al.'s study (2006), a NIH-funded project that represents one of the most significant research in this field. Although Cohen et al. used the bag-of-words method to extract the features and did not employ any re-sampling methods, these two studies are comparable since we used the same datasets, the same data sources (including titles, abstracts, MeSH, and MEDLINE publication type) in each dataset to extract features, and the same evaluation metrics (including precision, recall, F1 score and WSS@95%). Figure 1 shows the comparison of our proposed method with the benchmark model.

As shown in Figure 1, our approach that includes a combination of different text analytics techniques produced higher recalls, precisions, and F1-scores over all four datasets, compared with the benchmark model. The significantly improved WSS values indicate that our approach significantly reduced the number of articles that scientists need to manually review to develop systematic reviews, thereby having the potential to reduce labor and other costs associated with systematic reviews. Our proposed approach worked especially well for the datasets AN and SKE, each of which has only a few included article. For example, our approach produced 72.87% WSS@95% for the dataset SKE.

Reviewers initially queried and included 809 documents in the dataset SKE. A manual process will entail reviewing all 630 documents to end up with five relevant documents. In contrast, our proposed approach would have accurately removed 174 documents. This leaves only 152 articles for the reviewers to manually review (resulting in the five relevant articles).

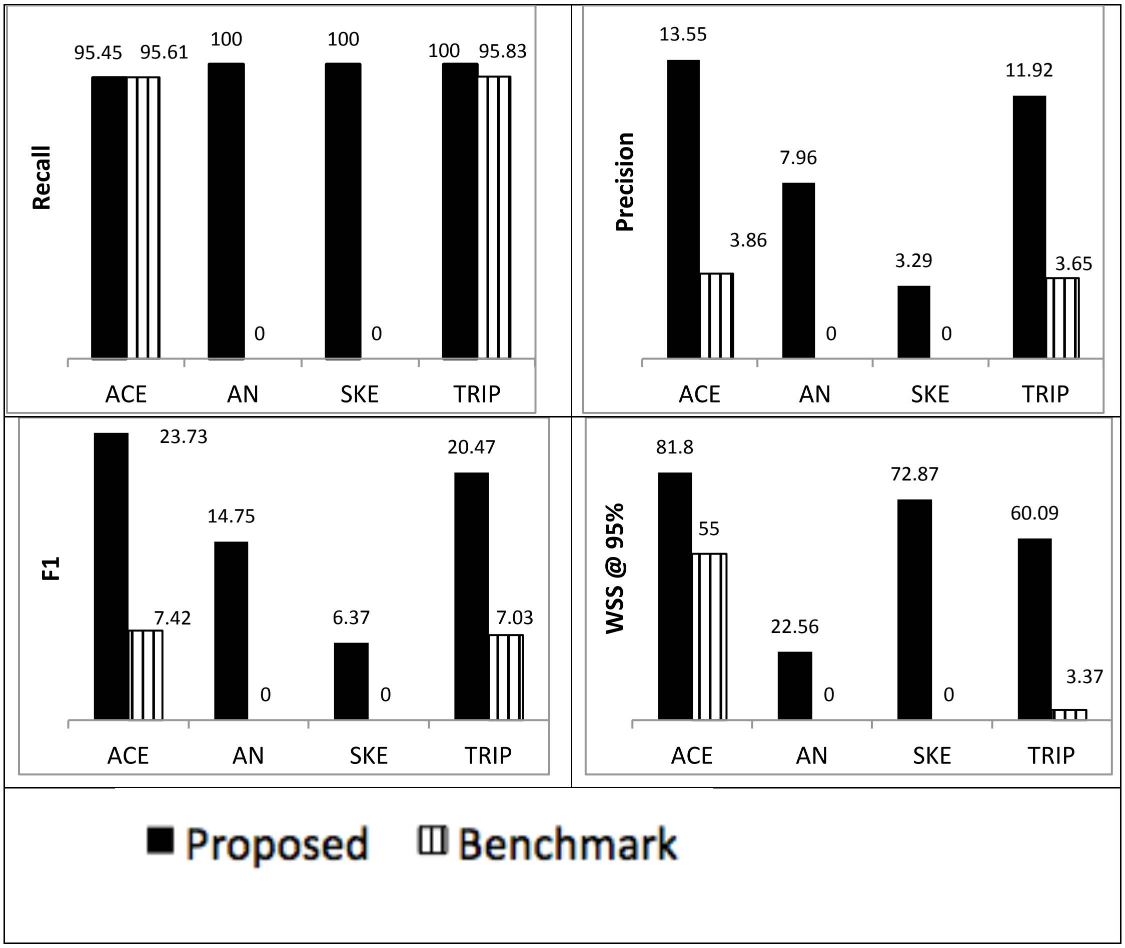


Figure 1: Comparison of proposed model with the benchmark model—Systematic Review Update

2.6. Conclusion

In this section, we examined an automated method to classify relevant articles for inclusion or exclusion during the abstract triage stage for updating systematic reviews of medical research. We demonstrated that a novel combination of text analytics techniques, including using the automatically extracted UMLS terms as the features, soft-margin polynomial SVM as the classification algorithm and SMOTE combined with undersampling to deal with the class balance issue, help improve precision while sustaining a high recall (95% or higher) in article classification for SRs.

3. Systematic Review Creation

3.1. Introduction

The general procedure of creating a new medical systematic review is similar to systematic review update; however, new challenge emerges because a training dataset is not available and researchers are attempting to answer completely new medical questions. The current workflow for creating systematic reviews is largely a manual process. It consists of 1) performing keyword search to identify potentially relevant articles, 2) performing article triage to identify articles for inclusion, and 3) finally, summarizing the selected studies via meta-analysis or other review methods. Within the workflow, article triage - identifying articles for inclusion in a systemic review - is particularly resource intensive (Shojania et al. 2007).

In that regard, various machine learning methods have been proposed to automate the article screening for systematic reviews (Bekhuis and Demner-Fushman 2012; Shemilt et al. 2013; Adeva et al. 2014). Supervised learning has has proven helpful during "abstract triage", where the abstracts of thousands or tens of thousands of articles retrieved from medical databases are reviewed and classified into "relevant" and "irrelevant". Supervised learning assumes a readily available training dataset. For instance, Cohen et al. (2006b) proposed a perceptron-based classifier that helps automatically identify relevant articles. The corpus used in the study includes 24 datasets on different medical topics collected by scientists at the Oregon Evidence-based Practice Center for the Drug Effectiveness Review Project (DERP). DERP scientists labeled each article in the dataset as "relevant" or "irrelevant" based on the abstract alone. Only the relevant articles would be included in full-text triage. When comparing various supervised learning algorithms for article selection, Adeva et al. (2014) used a dataset called Internet-Based Randomized Control Trial (IBRCT) mapping. It consists of 1941 articles that were read and classified by a committee of experts into 510 relevant and 1431 irrelevant instances. Supervised learning relies on a large training dataset, which can be problematic in this context because when we create a new systematic review, training data is rarely available. Cohen et al. (2006b) admitted the problem and focused on predicting which new articles are most likely to include evidence warranting inclusion in a review update. According to Cohen et al. (A. M. Cohen et al. 2009), the procedures for creating and updating systematic reviews (SRs) are similar; however, one important difference is that an SR update already has a collection of included/excluded article judgments that are based on previous reviews. Due to the lack of considerable amounts of training data, supervised learning methods proposed in exiting research holds very little promise for systematic review creation. Given a medical problem, a keyword search can return thousands or tens thousands of articles. Labeling these articles to create a sufficiently large training dataset is difficult, laborious and time-consuming. Scientists can afford to create a small-sized training set. However, it is known that a small-sized training dataset often leads to an overly simple prediction function that may not be rich enough to capture the true underlying relationship.

In recent years, semi-supervised has received considerable attention in the area of data mining due to its potential for reducing the effort of labeling data. Semi-supervised learning falls between supervised and unsupervised learning techniques. It refers to the method of using a large unlabeled data set *U* together with a given labeled dataset *L* in order to generate prediction functions that are more accurate on new data than would have been achieved using just *L* alone. It is motivated by the fact that in many settings, unlabeled data is plentiful but labeled data is limited or expensive. When it comes to creating a new systematic review, labeled training data (i.e., articles that have been reviewed by human experts) is mostly not readily available and is costly to obtain, requiring a manual review of thousands of articles. The goal of this research therefore is to perform an exploratory analysis of semi-supervised learning techniques for article selection for medical systematic reviews. More specifically, we plan to explore the ability of semi-supervised learning to overcome the labeling bottleneck and automate systematic review creation with a small-sized training dataset that includes, say, one or two hundred labeled articles. We perform comparative studies of various semi-supervised learning methods and identify the techniques suited for systematic review creation. To our knowledge, the proposed research is one of the first that conducts a comprehensive study on the feasibility of using semi-supervised learning to address the small-sized training dataset problem that hampers the use of classification algorithms for systematic review creation.

3.1.1. Related Work

Nowadays, there are public databases such as a global network of Cochrane entities and a North American network of AHRQ-funded Evidence-based Practice Centers that enables scientists to access up-to-date research findings. Even so, developing a systematic review is slow. The average time to complete a systematic review is 2.4 years with a reported maximum of 9 years. A bottleneck occurs during "abstract triage", where scientists screen the title and abstract of thousands or tens of thousands of articles for inclusion in a systematic review. Hence, most of the existing research has focused on automating abstract triage using supervised learning methods (Ananiadou et al. 2009; Bekhuis and Demner-Fushman 2012; A. Cohen et al. ; Frunza et al. 2010; Shemilt et al. 2013). Cohen et al. (2006b), in a National Institute of Health (NIH) supported project, developed a perceptron-based classifiers to identify journal articles for inclusion in systematic review update, based on the title and abstract of the articles. In another study, Frunza et al. (2010) applied naïve Bayes to a dataset of 47,274 manually labeled article abstracts. They obtained very high recall values (up to 99%) and moderately high precision of 63%. There are also studies that focus on comparing different algorithms that can be used to classify articles for systematic reviews. For instance, Bekhuis and Demner-Fushman (2012) compared different supervised learning algorithms including K-nearest neighbor (K-NN), naïve Bayes, complement naïve Bayes (cNB), and evolutionary SVM (EvoSVM) for "abstract triage". The authors demonstrated that based on text mining techniques, the number of documents that need to be further manually screened was reduced by up to 46%, and among the three algorithms, EvoSVM achieved the highest recall (100% for both datasets) and relatively low precisions (13.11% for the ameloblastoma dataset and 10.69% for the influenza dataset). Timsina et al. (2015) compared different supervised algorithms including SVM, Naïve Bayes, perceptron, etc., exploited Unified Medical Language Systems (UMLS) for medical terms extraction, and examined various techniques to resolve class imbalance issues. Through an empirical study, they demonstrated that SVM with polynomial kernel achieves better classification performance than other existing algorithms, and the performance of the classifier can be further improved by exploiting UMLS to identify medical terms in articles and applying re-sampling methods to resolve the class imbalance issue. Adeva et al.'s (Adeva et al. 2014) conducted experiments that involved multiple classification supervised learning algorithms (including naïve Bayes, k-Nearest neighbor, Support vector machines, and Rocchio) combined with several feature selection methods (including TF, DF, IDF, etc.), and applied to different parts of the given articles (including titles alone, abstracts alone and both titles and abstract). SVM has produced the highest F-measure when applied to the titles/abstracts. All these studies developed supervised learning classifiers based on large training datasets with manually designated labels. As discussed previously, a conspicuous problem with the supervised learning based approach to article selection is that supervised learning, to be effective, requires large amounts of training data, which is often not readily available in most circumstances when we create a new systematic review. It is time-consuming and resource-intensive for scientists to screen thousands of articles (even just the title/abstract of the article) to create a training dataset. In view of the problem, Cohen et al. (2006) suggested to focus on updating a review, where a reviewer already has a set of relevant documents in the form of the studies included in the original review.

Is it possible to develop a new systematic review without asking scientists to manually review thousands of articles? There are a few studies that attempted to provide feasible solutions to the problem. Cohen et al (2009) investigated whether a topic-specific automated document ranking system for systematic reviews (SRs) can be improved using a hybrid approach, combining topic-specific training data with data from other SR topics. The authors found that when topic-specific training data are scarce, leveraging training data previously used for developing systematic reviews for other related topics can significantly enhance the classification performance. There is also research that focuses on prioritizing the order in which citations (including titles, articles, keywords, etc.) will be screened. Tomas et al. (2011) suggested a possible method called “term recognition”, which works by treating the included titles and abstracts as one big (and growing) document. This method can start with a relatively small number labeled articles. Each time another article is marked as “included”, its text is added to the previously included titles and abstracts. The key terms from this string of text are then identified, and a search is carried out on the remaining titles and abstracts. The search is weighted by the significance attached to each term and the results ordered in terms of relevance. Thus, rather than viewing the documents in no particular order, those most similar to the studies already included are reviewed first. Unfortunately, no empirical results were presented on this “term recognition” method.

3.1.2. Research Gap

Overall, the findings of extant research indicate that supervised learning shows enough promise for automating the article selection process for systematic reviews if sufficient training instances are available. This is however a big “if” since developing a sufficiently large training set often requires screening the title/abstract of thousands of articles. Extensively studied in machine learning and applied to text classification, semi-supervised learning has been proved to be effective in case of a small-sized training dataset (e.g., Song et al., (2011); Jin, (2011)). Nonetheless, little research to date has examined if semi-supervised learning can help truncate the costly and laborious article screening process for systematic reviews by requiring a small percentage of labeled instances. This leads us to the following research questions:

1- *Is semi-supervised learning a viable technique for systematic review creation with limited labeled articles?*

To address this issue, we compare semi-supervised learning techniques with supervised learning to determine if semi-supervised learning produces more meaningful empirical results when used with a small-sized training dataset.

2- *Which semi-supervised learning method is most valuable for article selection for systematic reviews?*

We compare the performance of different semi-supervised learning algorithms and then investigate if combining "self-training" and "active-learning" with the best performing algorithm can further enhance article classification performance.

3.2. Article Classification

Our study includes three major components: 1) comparing the classification performance of different semi-supervised learning algorithms for systematic review article selection; 2) determining if combining "self-training" with the best performing algorithms identified in the previous step helps enhance classification performance, and 3) determining if combining "active-learning" with the best performing algorithms helps enhance classification performance. We conducted three experiments using three systematic review datasets. Before we describe our experiments in detail, we first describe the data sources, the semi-supervised methods, and the evaluation metrics for article classification used in our research.

3.2.1. Datasets and data processing

We used three systematic review datasets on drug topics including AtypicalAntipsychotics (AT), NSAID, and Estrogens (ESTRO) collected by AHRQ's

Evidence-based Practice Center (EPC) at Oregon Health and Science University in our research. These three systematic review datasets were also used in (A. Cohen et al. 2006). We wanted to identify if our experiments consistently produce desirable results across multiple sample sets; therefore, we used different datasets for SR creation and SR update procedure. Table 11 shows an overview of the datasets. Since, class imbalance issue is norm of systematic review dataset, Table 11 shows that there are much more irrelevant articles than relevant ones in all three datasets.

Table 11: Overview of Datasets—Systematic Review Creation

Dataset	Total number of articles	Number of articles labeled as relevant	Number of articles labeled as irrelevant	Ratio— relevant vs. irrelevant
Antihistamines (AT)	1120	363	757	0.48
Estrogens (ESTRO)	370	81	289	0.28
NSAID	393	88	305	0.29

Each document in our datasets includes the title, abstract, Medline publication type, and Medical Subject Heading of an article. After stop-word removal and stemming, we treated each document as a "bag of words", and each document was represented by a vector consisting of the TF-IDF weights of the words. TF-IDF is a numerical statistic that reveals the importance of a word to a document in a dataset. The TF-IDF value of a word increases as it appears more often in a document, but is offset by the frequency of the word in the whole dataset. This helps to mitigate for the fact that some words such as "patient" and "disease" are generally more common than other words in medical documents.

3.2.2. Semi-supervised Learning Methods

We investigate the following semi-supervised learning methods.

Label Spreading (Zhou et al. 2004): Label Spreading assumes that geometrically closer data points tend to be similar. There are two general ideas related to label spreading: 1) the labeled examples act as sources that push out labels to unlabeled data, and 2) an example propagates its label to its neighboring examples according to their proximity to it.

Formally, we are presented with a set of n data points $X = X_{\mathrm{L}} \cup X_{\mathrm{U}} = \{x_1, ..., x_l, x_{l+1}, ..., x_n\}$, where X_{L} represents the labelled subset, X_{L} the unlabeled subset, and $x_i \in \mathrm{R}^m$ is a m-dimensional feature vector representing the i-th data point. For the first l data points, we have the corresponding labels $Y_L = \{y_1, ..., y_l\}$, where $y_i \in \{-1, 1\}$. Let $F(0) = \{ y_1, ..., y_l, 0, ..., 0\}$ be the vector that represents the labels for the data points, where $y_i = 0$ for $l < i \leq n$.

We code the data as a graph $G = (X, W)$. The nodes X represent individual data points, and the edges are coded in an affinity matrix W. W stores the similarity between data points. In our research, we used the RBF kernel (radial basis function) as the similarity function, i.e., $W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)\ \ \forall i \neq j$, and $W_{ii} = 0$. The normalized graph Laplacian L is defined as

$$L = D^{-1/2}WD^{-1/2} \text{ with } D_{ii} = \sum_j w_{ij} \ .$$

Label spreading proposed in is a graph based semi-supervised learning technique that spreads the label information from X_{L} to the X_{U} based on the affinity of the data points. It does this via the normalized graph Laplacian, and mathematically, the iterative information spreading is:

$$F(t + 1) = \alpha LF(t) + (1 - \alpha)F(0),$$

where α is a suitably chosen learning rate. This iteration converges to the solution

$$F* = (1 - \alpha)(I - \alpha L)^{-1}F(0),$$

The solution $F*$ can be interpreted as node weights after Y_L has been propagated across the graph. Due to the smoothness constraints, reliable labels should reinforce each other, resulting in higher node weights, whereas labels showing inconsistencies tend to cancel out, resulting in lower node weights.

Label Propagation (Zhu and Ghahramani 2002): Label propagation is similar to Label spreading in that both algorithms are graph-based, and both attempt to propagate a node's label to its neighboring nodes according to their proximity. Using label propagation, we also construct the graph $G = (X, W)$, where X represents individual data points, and the edges are coded in an affinity matrix W. The major difference between label propagation and label spreading is that label propagation uses the raw similarity matrix W constructed from the data with no modifications, while label spreading iterates on a modified version of the original graph and normalizes the edge weights by computing the normalized graph Laplacian matrix L (see above).

Semi-supervised Support Vector Machine (S3VM) (Bennett and Demiriz 1999): S3VM, an extension of standard support vector machine with unlabeled samples, is another widely used semi-supervised learning technique. The goal of an S3VM classifier is to find a labeling of the unlabeled samples, so that a linear boundary has the maximum margin on both the original labeled samples and the (now labeled) unlabeled samples. The obtained decision boundary has the smallest generalization error bound on unlabeled samples.

Formally, standard linear supervised SVMs output a decision function of the form *f(x)* = sign ($w^Tx + b$), by minimizing the following objective function

$$\frac{1}{2}\ \mathrm{w}^T\mathrm{w}\ +\ \mathrm{C}\ \sum_{i=1}^{n} L(y_i(w^T x_i + b))\ ,$$

where l (t) = max(0, 1−t) is a hinge loss function penalizing the training errors and C is a trade-off constant.

In the semi-supervised case, an additional term is added to the objective function that drives the outputs $w^T x_i$ + b of the unlabeled point x_i away from 0 (thereby implementing the cluster assumption):

$$\frac{1}{2}\ \mathrm{w}^T\mathrm{w}\ +\ \mathrm{C}\ \sum_{i=1}^{l} L\big(y_i(w^T x_i + b)\big) +\ \mathrm{C}^*\sum_{i=l+1}^{n} L(|w^T x_i + b|)$$

The main problem is that this additional term in the objective function (5) is non-convex, which make optimization difficult (Zhu 2005).

We selected the above three semi-supervised learning algorithms because they are widely used, and we have reliable implementations of them. Scikit-learn, a well-known machine learning toolkit, includes implementations of label spreading and label propagation. We used the S3VM implementation developed by (Gieseke et al. 2014).

In our research, we also considered two wrapper methods for semi-supervised learning: Self-training and Active Learning. They are wrapper methods because they "wrap" some existing classifiers. In self-training, an existing classifier (such as SVM) is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically, the most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated.

Active learning is a special type of semi-supervised learning. Active learning resembles self-training in that it also attempts to overcome the labeling bottleneck by identifying the most informative set of unlabeled instances based on some existing classifiers. It differs from self-training in that after selecting the most confident unlabeled samples, it requests an oracle (e.g., a human expert) to assign their labels. Active learning is also an iterative process in which it first trains a classifier with few training instances, based on the training results, it selects an optimal set of unlabeled instances and queries an oracle for manual labeling, and then it re-trains the algorithm based on the incremented training data.

3.3. Evaluation

We evaluated article classification performance using the classical precision, recall, and F1 metrics. The formulas for computing recall, precision, and F1are shown in Table 12. TP represents the number of True Positives, i.e., positive samples that were correctly classified. TN is the number of True Negatives, i.e., negative samples that were correctly classified, FP the number of False Positive, i.e., negative samples that were incorrectly classified as positive, and FN the number of False Negatives, i.e., positive samples incorrectly classified as negatives. Recall refers to the rate of correctly classified positives among all positives and is equal to TP divided by the sum of TP and FN. Precision refers to the rate of correctly classified positives among all examples classified as positive and is equal to the ratio of TP to the sum of TP and FP. F1 represents the harmonic mean of recall and precision.

Table 12: Evaluation metrics—Systematic Review Creation	
Evaluation Metric	Formula
Recall	TP/ (TP+FN)
Precision	TP/(TP+FP)
F1	(2*recall*precision)/(recall + precision)

3.4. Experiments

We conducted three experiments to evaluate the effectiveness of the various semi-supervised learning methods for article selection for systematic reviews. The datasets we used in the experiments are the three datasets we described in Table 11.

3.4.1. Experiment 1 – Comparing different semi-supervised learning algorithms

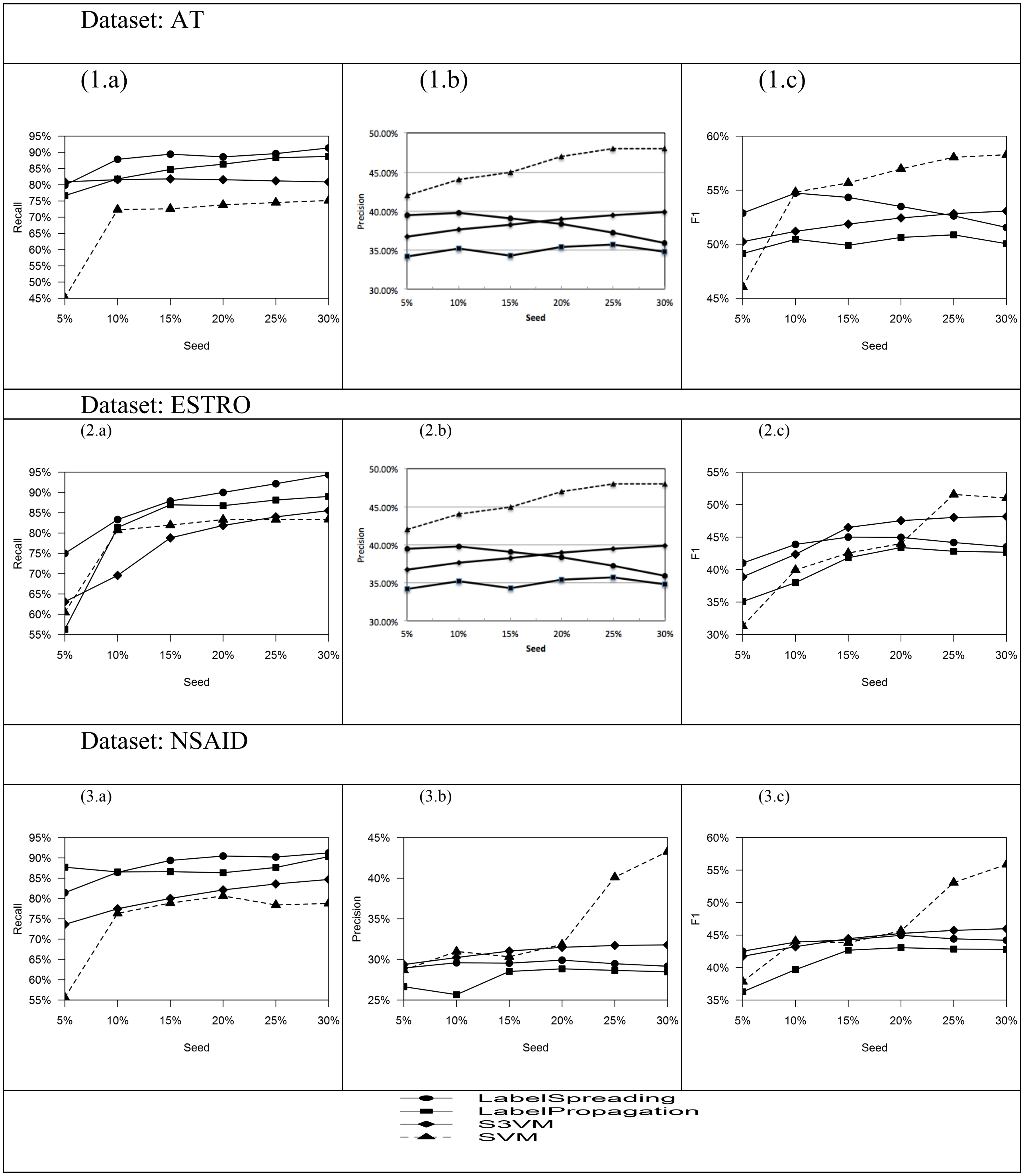
In Experiment 1, we evaluated the effectiveness of three generic semi-supervised learning algorithms including label spreading, label propagation, and S3VM. We compared the performance of these semi-supervised learning algorithms with standard supervised SVM with polynomial kernel. SVM with polynomial kernel has been proved to achieve better performance than others in a recent study that compares a variety of supervised learning algorithms for article selection for systematic reviews (Timsina et al. 2015).

a. Experiment design

We started with 5% labeled articles as seeds or initial training instances. We conducted stratified sampling to make sure that 5% of the positive instances and 5% of the negative instances in the seeds. Using the 5% seeds (i.e., initial labeled instances) as the training set and the rest 95% samples as the test set, we conducted semi-supervised learning using the three different algorithms. Since the seeds were randomly sampled, this random sampling would have a substantial effect on the performance of the classifiers. Hence, for each algorithm, we conducted 50 trials to ensure the reliability of the results. We started with label spreading. In each trial, we first randomly selected 5% seeds including 5% of the positive instances and 5% of the negative instances are in the seeds and then performed learning. We then averaged the results of 50 trials to generate the final results for the label spreading algorithm with 5% seeds. This approach is consistent with an earlier approach used in literature (Zhu and Ghahramani 2002). For the other algorithms including label propagation, S3VM, supervised SVM, we did not re-select the seeds. Rather, we used the 5% seeds that were previously selected in the 50 trials for label spreading to ensure that we compared the different algorithms using the same training and test sets. After getting the results with 5% seeds, we increased the number of seeds to 10%, 15%, 20%, 25%, and 30%. For each number of seeds, we again conducted 50 trials and obtained the average results.

b. Results and findings

The results of Experiment 1 are shown in Figure 2.



Dataset: AT

(1.a)

(1.b)

(1.c)

Dataset: ESTRO

(2.a)

(2.b)

(2.c)

Dataset: NSAID

(3.a)

(3.b)

(3.c)

Figure 2. Experiment 1 Results—Systematic Review Creation

Among the three measures including recall, precision and F1, recall is probably the most important one in this context. Any automated system for identifying relevant articles must maintain a very high level of recall since ideally, a systematic review should include all articles that provide high quality evidence relevant to a topic. Any system with a low recall would be of little use (Matwin et al. 2010). Cohen et al. (2006) even assumed that a recall of about 0.95 is required for a classification system to identify an adequate fraction of the positive papers. The diagrams (1.a), (2.a) and (3.a) in Figure 2, respectively, show the recall results when we applied the three semi-supervised learning algorithms and the benchmark supervised SVM to the three datasets AtypicalAntipsychotics (AT), Estrogens (ESTRO), and NSAID. Label spreading consistently achieved higher recall than the other algorithms across all three datasets. When applied to the dataset AT, label spreading obtained around 90% recall with over 10% seeds. For the dataset ESTRO, label spreading produced recall of 83.32% with 10% seeds and raised recall to 90% with 20% seeds and to 94.36% with 30% seeds. It also produced recall of 90.32% with 20% seeds and recall of 91.23% with 30% seeds for the dataset NSAID. Label propagation also achieved relatively high recalls for all three datasets. In two datasets, AT and NSAID, with 30% seeds, label propagation and label spreading produced similar recall results. However, label spreading consistently achieved higher recall than label propagation when the number of seeds is smaller than 25%. S3VM and SVM produced lower recall results than the two graph-based algorithms including label spreading and label propagation. S3VM produced recall results similar to those obtained by standard supervised SVM for two datasets ESTRO and NSAID with over 10% seeds. It appeared that S3VM failed to produce a recall that is high enough to make it a feasible method for article selection with a small-sized training set. The highest recall values yielded by S3VM for the three datasets include 81.81 % for AT, 85.52% for ESTRO, and 84.69% for NSAID.

The diagrams (1.b), (2.b) and (3.b) in Figure 2 display the precision results obtained by different algorithms for the three datasets. Precision is still essential in this context, but it is only meaningful when a high recall has been achieved. A higher precision means that the articles that are classified as "relevant" are indeed relevant, which means that a smaller number of articles needed to be manually reviewed. The diagram (1.c), (2.c) and (3.c) show the F1 scores. In this area, F1 is not as important a measure as it is in other contexts. F1 represents the harmonic mean of precision and recall. It hence assumes equal misclassification costs for false positive and false negative errors, but in the context of article selection for systematic reviews, an error of missing a relevant article (i.e., a false negative error) can be more expensive than an error of selecting an irrelevant article (i.e., a false positive error). After all, the articles selected by machine learning methods still need to be manually verified. Among the three semi-supervised algorithms, S3VM consistently achieved higher precision results than label spreading and label propagation. Also, compared with label spreading, S3VM produced similar F1 scores to label spreading for two datasets (AT and NSAID) and higher F1 scores (46.49% vs.45.00% with 15% seeds, and 48.17 % vs. 43.50% with 30% seeds) for ESTRO. With respect to the metrics precision and F1, supervised SVM performed even better than S3VM. For the dataset AT, it yielded over 47% precision, roughly 10% higher than those obtained by S3VM and label spreading. For the other two datasets (ESTRO and NSAID), the precision results and subsequently F1 scores obtained by supervised

SVM underwent a jump between 20% seeds and 25% seeds, indicating that a supervised learning algorithm such as SVM requires a certain number of training instances (more than 20% in this case) to take effect. Such a jump, however, did not occur to SVM's recall results. Even with 30% seeds, SVM produced low recall results (76.99% for AT, 83.32% for ESTRO, and 78.79% for NSAID)

In summary, the two graph-based semi-supervised methods, label spreading and label propagation, produced higher recall results than S3VM and SVM, while S3VM and SVM (with more than 20% seeds) produced relatively higher precision results. It appeared that the graph-based methods and the SVM-based algorithms have both advantages and disadvantages. Further analysis showed that label spreading and label propagation produced a significantly larger number of true positives than S3VM and SVM, which means label spreading and label propagation were able to identify some positive instances (i.e., relevant articles) that were missed by S3VM and SVM. With a significantly larger number of true positives, label spreading and label propagation achieved higher recall values. On the other hand, label spreading and label propagation also made a significantly larger number of false positive errors than S3VM and SVM. A false positive error means that a negative instance (i.e., an irrelevant article) was falsely classified as positive (i.e., relevant). As a result, overall, label spreading and label propagation yielded a lower level of precision than S3VM and SVM. In the context of systematic reviews, high recall is a prioritized criterion for effective article classification algorithms. Precision is useful only when a high level of recall is obtained. We hence believe that in this context, the graph-based algorithms are preferred to the SVM based algorithms. Between the two graph-based methods, label spreading performed better than label propagation with respect to both recall and precision. A plausible reason can be label spreading minimizes a loss function that has regularization properties, as such it is often more robust to noise. Hence, among the three semi-supervised learning methods we investigated, label spreading appeared to be the optimal method for dealing with article selection for systematic reviews with limited labeled instance. Moreover, the results of experiment 1 indicates that semi-supervised learning, more specifically label spreading, could be a viable method for systematic review article selection with limited labeled instances. Label spreading obtained high recall in all three datasets. It achieved over 90% recall for AT and NSAID and about 95% recall for ESTRO. However, if we follow Cohen et al.'s requirement that a recall close to 95% is imperative for classification algorithms, further improving recall necessary. It is also noteworthy that compared with standardized supervised SVM, label spreading produced lower precision results. Although not as critical as recall in this context, lower precision signifies more false positive errors, which means that more irrelevant articles would be manually reviewed. We hence conducted the next two experiments, Experiment 2 and Experiment 3, to explore methods for further enhancing classification performance.

3.4.2. Experiment 2 – Enhancing classification performance with self-training

The goal of this experiment is to investigate if combining label spreading with self-training and supervised SVM can improve precision while maintaining or even enhancing recall, thus helping further reduce workload for systematic review article selection. Self-training is a semi-supervised method that can be used to increment the training set. Given an initial training dataset, self-training relies on an existing algorithm to label some of the most confident unlabeled instances. It then adds the newly labeled instances to the training dataset and re-trains the algorithm. This process can be iterated over the remaining unlabeled data. Supervised learning algorithms such as SVM have often been used in self-learning to identify the most confident instances. In this experiment, we used label spreading, a semi-supervised algorithm, to select the optimal unlabeled instances since in Experiment 1, label spreading produced much higher recall and identified more true positives than SVM with a small-sized training dataset.

a. Experiment design

We used different numbers of seeds (i.e., initially labeled articles) ranging from 5% to 30%. Again, to alleviate the effect of random sampling, for each seed number, we conducted 50 trials. Using the seeds as the initial training dataset, we performed label spreading learning to classify the unlabeled instances. Label spreading computed a weight for each unlabeled instance. An unlabeled instance with a higher weight was considered more likely to be positive. We ranked the unlabeled instances according the weights that the label spreading method produced for them. We then selected a few top instances and a few bottom ones and incorporated them with their predicted labels into the training set. This completed one iteration of self-training. The incremented training dataset was used to re-train the label spreading algorithm in the next iteration. Among the three datasets, ESTRO and NSAID have similar numbers of positive instances and negative instances. We tested different numbers of iterations (from 4 to 12) for these two datasets, and in each iteration, we also tried to select from 10 (including top 5 and bottom 5 instances) to 20 instances (including top 10 and bottom 10 instances). It appeared that 9 iterations of self-training with top 8 and bottom 8 instances selected in each iteration produced the best performance for ESTRO and NSAID. The dataset AT has a much larger number of positive and negative instances. We hence conducted 18 iterations of self-training with top 8 and bottom 8 instances selected in each iteration to make sure that relatively similar percentages of new instances would be labeled and added to the training set across all three datasets. Table 13 includes a column called “Final Train”, which shows the final sizes of the incremented training datasets after the iterative self-training process. For instance, for the dataset AT, the initial training set included just the 5% seeds. 18 iterations of self-training added 40.44% instances to the training set, which resulted in a final training dataset that included 45.44% (40.44% new instances + 5% seeds) of the total instances. With the final training set, we trained a supervised SVM classifier and classified the remaining unlabeled instances.

Results and findings

Table 13: Experiment 2 results—Systematic Review Creation

Dataset	Seed	Self-training				Label Spreading			SVM		
		Final Train*	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
AT	5%	45.44%	80.26%	45.26%	57.88%	79.87%	39.52%	52.88%	45.18%	46.98%	46.06%
	10%	52.84%	85.58%	45.08%	58.81%	84.85%	39.75%	54.73%	72.36%	44.40%	55.03%
	15%	60.24%	85.42%	45.42%	59.31%	88.63%	39.02%	54.33%	72.59%	45.25%	55.75%
	20%	67.64%	87.33%	45.33%	59.68%	88.59%	38.31%	53.49%	73.80%	46.54%	57.08%
	25%	75.03%	88.89%	44.89%	59.65%	89.61%	37.25%	52.62%	74.52%	47.69%	58.16%
	30%	82.43%	90.17%	45.34%	60.34%	91.34%	35.91%	51.55%	75.14%	47.76%	58.40%
ESTRO	5%	54.67%	86.74%	29.35%	43.86%	74.99%	28.45%	41.01%	60.48%	21.14%	31.33%
	10%	59.86%	89.22%	30.93%	45.94%	83.32%	29.98%	43.88%	80.72%	26.54%	39.94%
	15%	64.71%	90.87%	35.64%	51.20%	87.88%	30.41%	45.00%	81.94%	28.74%	42.55%
	20%	69.90%	89.53%	36.44%	51.80%	90.00%	30.11%	44.96%	83.33%	29.89%	44.00%
	25%	74.74%	92.16%	38.38%	54.24%	92.15%	29.17%	44.17%	83.32%	37.34%	51.57%
	30%	79.93%	93.75%	38.66%	54.74%	94.36%	28.38%	43.50%	83.36%	36.74%	51.01%
NSAID	5%	52.13%	82.60%	30.35%	44.40%	81.43%	28.94%	42.51%	55.63%	29.37%	38.44%
	10%	57.38%	86.37%	30.93%	45.41%	86.45%	29.59%	43.92%	76.35%	30.98%	44.07%
	15%	62.30%	86.17%	32.64%	47.35%	89.38%	29.54%	44.25%	78.88%	30.33%	43.82%
	20%	67.21%	89.11%	38.44%	53.71%	90.46%	29.91%	44.96%	80.64%	31.84%	45.65%
	25%	72.13%	89.56%	43.38%	58.45%	90.22%	29.47%	44.43%	78.41%	40.12%	53.08%
	30%	77.38%	90.78%	43.66%	58.97%	91.23%	29.16%	44.19%	78.79%	43.28%	55.87%

Note: * “Final Train” stands for final training set size. Self-training added new labeled instances to the training set. This field indicates the size of the final training data size after the iterative self-training

We compared the performance of self-training with that of using label spreading alone and of using SVM. Table 13 shows the results, with the largest recall, precision and F1 scores for each dataset with a specific number of seeds being highlighted.

We were aware that existing studies such as (A. Cohen et al. 2006; Bekhuis and Demner-Fushman 2012) had shown a tendency for recall to decline when precision increases. Since the Experiment 1 results showed that supervised SVM achieved lower recall but higher precision than label spreading, we decided to use SVM to train a portion of the unlabeled instances, which could potentially enhance precision but lower recall. We attempted to remedy this by using self-training to increment the training dataset. Our strategy hence included using self-training to increment the training set, in order to maintain a high level of recall, and using the incremented training set to train a supervised SVM learner, in order to enhance precision. Obviously, our strategy has been proved to be effective in enhancing precision in Experiment 2. Compared with label spreading, self-training produced significantly higher precision for all three datasets. For instance, for the dataset AT and ESTRO, self-training with 30% seeds produced precision that is about 10% higher than the precision obtained by label spreading alone. For the dataset NSAID, self-training with 30% seeds produced precision of 43.66%, while label spreading with the same seeds produced precision of only 29.16%. Self-training also produced very comparable precision results to SVM. Our strategy was also effective in maintaining a high level of recall. It worked especially well with a small number of seeds. For the dataset AT, with 5% and 10% seeds, self-training achieved higher recall (80.26% vs. 79.87% for 5% seeds and 85.58% vs. 84.85) than label spreading alone. For ESTRO with 5%, 10%, and 15% seeds and for NSAID with 5% seeds, self-training also yielded slightly higher recall. When the number of seeds got larger, self-training obtained slightly lower recall than label spreading alone.

To summarize, in Experiment 2, we aimed to enhance precision while maintaining or, better, improving recall. We used self-training with label spreading to identify the most confident unlabeled instances. These instances with their predicted labels were incorporated into the training dataset, and with the incremented training set, we employed SVM to classify the remaining unlabeled instances. The self-learning method succeeded in enhancing precision and maintaining a high level of recall. It, however, failed to further enhance recall. A reason could be that even if we chose to add the most confident instances in self-learning, some instances were still misclassified. In Experiment 2, across the three datasets, we labeled 1800 unlabeled instances as positive. We made 177 (or 9.83%) false positive errors. Our self-training method was much more effective in identifying negative instances, probably because our datasets are imbalanced, i.e., there are far fewer "relevant" than "irrelevant" instances in all three datasets. Among 1800 instances labeled as negative in the self-training process only 28 (1.56%) were misclassified. A serious limitation of self-training is that these misclassified instances were treated as truth and were used to classify other unlabeled instances. The impact of these misclassified instances could snowball as the self-training process proceeded. We hence continued to explore the effectiveness of active learning. We expected that with human labeled instances incorporated into the training dataset, we could enhance both recall and precision.

3.4.3. Experiment 3 – Enhancing classification performance with active learning

Active learning approach has received considerable attention due to its potential for achieving greater classification accuracy in applications where unlabeled data may be abundant or easily obtained, but labels are difficult, time-consuming, or expensive to obtain (Settles 2010). Active learning is similar to self-training in that the learner is responsible for acquiring training samples. The main difference of active learning from self-training is that in active learning, after an optimal set of unlabeled instances were identified, human experts need label these instances. In this experiment, we wanted to investigate whether active learning based on label spreading can further enhance the performance of article classification, as compared with the fully automated approaches such as the self-training method described above.

a. Experiment design

For each dataset, we again used different numbers of seeds. Again, to alleviate the effect of random sampling, given a specific number of seeds, we conducted 50 trials and took the average of the results. In each trial, we performed active learning iteratively. We conducted 9 iterations of active learning for the datasets NSAID and ESTRO and 17 iterations for the dataset AT. In each iteration, we added 6 articles predicted by the algorithm as negative and another 6 articles predicted as positive to the labeled set. We conducted multiple tests to identify these optimum parameters such as the number of iterations and the number of instances added to the training set. As discussed previously, there are more negative instance than positive ones in a typical systematic review dataset; machine learning hence tends to achieve high accuracy on predicting the negative articles, as evidenced by existing research (Shemilt et al. 2013). Our datasets indeed included much fewer "relevant" articles than "irrelevant" ones. The Experiment 2 results showed that label spreading is effective in identify negative instances, with only misclassified 1.56% negative instances. Thus, in our active learning method, we added those instances predicted by the label spreading algorithm as negative into the labeled set without asking human experts to annotate them. Positive articles, on the other hand, are fewer, and label spreading identified them with a higher misclassification rate in Experiment 2. In real practice, it is necessary for human experts to review and label the articles that were recognized as positive by label spreading and then add them to the training dataset. In our experiment, since the actual label of each instance is available in our datasets, we simply added the instances with their correct labels to the training dataset without asking human experts to review them. Like in Experiment 2, we used active learning to increment the training dataset iteratively. With the final incremented training set, we learned a SVM classifier, which was then used to classify the remaining unlabeled instance.

The sizes of the final training datasets after the iterative active learning process are shown in the column "Total Article Read" in Table 14 below. Each final training dataset after active learning included the initial seeds and the newly labeled instances. In real practice, both the seeds and the instances labeled during active learning represent manually reviewed instances. We used our self-training method and supervised SVM as the benchmark methods. We conducted self-training and supervised SVM classification with an initial training dataset that included the same number of instances as in the training set obtained by active learning. For instance, for the dataset AT with 5% seeds, the augmented training dataset after active learning encompassed 26.43% of the instances, which included 5% seeds plus 21.43% newly labeled articles – these are articles supposedly reviewed by human experts. When we conducted self-training using the method described in section 4.2 for comparison, we also created an initial training set that contained 26.43% instances (including 5% seeds and another 21.43% stratified samples). By doing this, we made sure that we compared actively learning and self-training based on an equal number of manually reviewed articles. We conducted supervised SVM classification using the same initial training set prepared for self-learning.

b. Results and findings

We compared the active learning method with supervised SVM and the self-training method described in section 4.2. Table 14 shows the comparison results.

Table 14: Experiment 3 results—Systematic Review Creation

Dataset	Seed	Total Article Read	Active Learning			Self-training			SVM		
			Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
AT	5%	26.43%	89.50%	50.54%	64.60%	88.98%	44.92%	59.70%	74.52%	47.69%	58.04%
	10%	31.43%	91.50%	49.52%	64.26%	91.17%	45.68%	60.86%	75.14%	47.76%	58.28%
	15%	36.43%	90.40%	51.12%	65.31%	90.17%	45.34%	60.34%	75.87%	47.78%	58.56%
	20%	41.43%	91.42%	51.53%	65.91%	90.76%	46.29%	61.31%	75.76%	47.66%	58.45%
	25%	46.43%	92.74%	52.26%	66.85%	91.37%	46.49%	61.62%	76.88%	48.22%	59.23%
	30%	51.43%	93.18%	54.81%	69.02%	91.92%	46.38%	61.65%	78.63%	47.73%	59.36%
	70%								89.95%	54.62%	66.98%
ESTRO	5%	27.84%	93.89%	41.82%	57.87%	92.87%	38.60%	54.53%	86.36%	37.34%	52.13%
	10%	32.70%	95.87%	42.49%	58.88%	93.97%	38.97%	55.09%	83.32%	36.74%	51.00%
	15%	37.84%	96.33%	42.67%	59.14%	94.22%	38.20%	54.36%	83.48%	36.69%	50.78%
	20%	42.70%	96.59%	41.68%	58.24%	94.69%	39.14%	55.39%	85.66%	36.93%	51.61%
	25%	47.84%	97.56%	41.95%	58.67%	95.16%	39.18%	55.50%	84.36%	37.19%	51.63%
	30%	52.70%	98.06%	44.77%	61.47%	95.64%	39.17%	55.57%	82.85%	39.88%	53.84%
	70%								93.38%	43.43%	59.28%
NSAID	5%	26.46%	91.60%	48.02%	63.01%	89.76%	44.02%	59.07%	78.41%	40.12%	53.08%
	10%	31.30%	92.94%	49.07%	64.23%	90.77%	44.53%	59.75%	78.79%	43.28%	55.87%
	15%	36.39%	93.34%	50.94%	65.91%	91.14%	45.16%	60.40%	77.97%	44.68%	56.81%
	20%	41.48%	94.01%	46.53%	62.25%	91.51%	45.61%	60.87%	77.10%	44.97%	56.80%
	25%	46.31%	94.44%	50.32%	65.66%	91.87%	45.37%	60.74%	78.40%	43.17%	55.68%
	30%	51.40%	94.90%	51.14%	66.46%	92.24%	46.53%	61.85%	79.02%	43.85%	56.40%
	70%								90.48%	49.51%	61.45%

Table 14 shows that the active learning method produced considerably better recall and precision than both self-training and supervised SVM. It worked well even with a small number of seeds. For instance, with 10% seeds (around 31% of total instances read), the active learning method produced recall of 91.50% for AT, of 95.87% for ESTRO and of 92.94% for NSAID. We also included SVM classification results with

70% training datasets in Table 14. Active learning consistently outperformed SVM with 70% training sets, with respect to all three metrics. They have comparable precision results. However, SVM, even with a large training set, still made quite some false negative errors and produced a level of recall that made it problematic to be used in the context of systematic review article selection. Another contributing factor that led active learning to outperform SVM could that be in each iteration of active learning, we selected roughly an equal number of positive vs. negative instances. In other words, the proposed active learning method implicitly performed under-sampling. Prem et al. (2015) proved that since a typical systematic review dataset includes much fewer relevant articles than irrelevant ones, employment of re-sampling methods dealing with class imbalance such as under-sampling can significantly improve the performance of machine learning classifiers.

In summary, we conducted three experiments, each of which shed some light on the use of semi-supervised learning in selecting articles for systematic reviews. The Experiment 1 results showed that given a small-sized training dataset, semi-supervised methods, especially label spreading, achieved a high level of recall, which makes them viable methods for reducing workload for systematic review article selection. The Experiment 1 results also showed that label spreading alone resulted in low precision. To improve precision while maintaining or better enhancing recall, we proposed a self-training based method that combines semi-supervised learning (with label spreading based self-training) and supervised learning (with SVM). The Experiment 2 results showed that the proposed self-training based method significantly enhanced precision while maintaining a high level of recall. It worked especially well with small training sets

(5% or 10% seeds). Next, we explored the feasibility of using active learning to further enhance both recall and precision. Experiment 3 results showed that active learning produced a very high level of recall that meets Cohen et al.'s 95% recall requirement, suggesting that the active learning method is a highly feasible method for systematic review article selection with small-sized training datasets. However, active learning requires human expert to be continuously engaged to produce optimum results. If experts' engagement is not available, with an initial small-sized training set, self-training provides a feasible alternative. It is fully automatic, though the classification performance of self-training is inferior to that of active learning.

3.5. Conclusion

We examined several different semi-supervised methods and identified label spreading as an algorithm that produced high recall that is necessary for systematic review article selection. We also demonstrated that the performance of label spreading could be further enhanced when we combined it with self-training and active learning.

4. Conclusion and Contribution

Evidence-based medicine has been widely promoted as a means of improving clinical outcomes, where evidence-based medicine refers to the practice of medicine based on the best available scientific evidence. Information overload, however, makes it difficult for healthcare providers to easily integrate evidence into practice. The challenge not only lie in recognizing the potential for breakthroughs in healthcare but in *realizing* this potential by providing the right tools to find the data that are relevant, extract information from the data, and convert that information to actionable knowledge. Information technology (IT) plays a crucial role in the practice of evidence-based medicine (EBM) by allowing health care practitioners to access and evaluate clinical evidence as they formulate their patient care strategies (Wells 2006). This oftentimes involves an analysis of a large amount of complex information.

This research focuses on systematic reviews, the heart of evidence-based medical practice (Stevens 2001). The creation and update of these reviews is resource intensive. A major bottleneck occurs when scientists screen medical studies. Scientists need to identify *provisionally eligible* studies by reading the title and abstract of thousands of articles. This challenge calls for the use of text analytics to automate the article selection process. Next we present our concluding remarks on usage of machine learning on systematic review update and creation.

4.1. Systematic Review Update

In this research, we examined an automated method to classify relevant articles for inclusion or exclusion during the abstract triage stage for updating systematic reviews of medical research. We demonstrated that a novel combination of text analytics techniques, including using the automatically extracted UMLS terms as the features, soft-margin polynomial SVM as the classification algorithm and SMOTE combined with undersampling to deal with the class balance issue, help improve precision while sustaining a high recall (95% or higher) in article classification for SRs. At first, we compared five algorithms (Soft-margin SVM, Perceptron, SVM, evoSVM and Naïve Bayes) with the features extracted using the bag-of-words approach plus the MeSH and MEDLINE publication type. Next, we compared those five algorithm with features including the automatically extracted UMLS terms plus the Medical Subject Headings (MeSH) and MEDLINE publication type. Our empirical investigation showed that 1) soft-margin polynomial SVM consistently performed better than the other algorithms across the four datasets, and 2) overall, using the UMLS terms as features helps enhance the performance of soft-margin polynomial SVM and the other algorithms as well. After demonstrating that soft-margin SVM is the better classification algorithm compared with the other algorithms in Experiment 1, we investigated if we can further enhance precision while maintaining a high recall using different re-sampling methods. We tested four re-sampling technique - undersampling, oversampling by replicating minority class examples, SMOTE, and SMOTE combined with undersampling. We demonstrated that 1) overall, SMOTE-based re-sampling methods including both SMOTE alone and SMOTE combined with undersampling helped improve classification performance of the soft-margin SVM classifier, whether we used the UMLS extracted features or bag-of-words; 2) the combination of SMOTE and undersampling in general performed better than SMOTE alone when the UMLS terms were used as the features.

4.2. Systematic Review Creation

This research presents a comprehensive study on the feasibility of using semi-supervised learning to select relevant articles for systematic reviews. Specifically, we examined label-spreading (various kernels), label-propagation (various kernels) and semi-supervised support vectors machines. Through empirical evidence, we identified label spreading as an algorithm that produced high recall that is necessary for systematic review article selection when the training dataset is small. We also performed comparison of semi-supervised based learning algorithm with supervised techniques. We concluded that semi-supervised based techniques outperforms supervised based techniques when the training dataset is smaller than 15-20 of total samples. Next, we investigated if combining label spreading with self-training and supervised SVM can improve precision while maintaining or even enhancing recall, thus helping further reduce workload for systematic review article selection. Here, we compared the performance of self-training with that of using label spreading alone and of using SVM. The self-learning method succeeded in enhancing precision and maintaining a high level of recall. It, however, failed to further enhance recall. We expected that with human labeled instances incorporated into the training dataset, we could enhance both recall and precision; thus, we investigated if active learning method, in which human labeled instances incorporated, can further optimize the result compared to self-learning and active learning. Active learning consistently outperformed SVM with 70% training sets, with respect to all three metrics. They have comparable precision results.

We conducted three experiments, each of which shed some light on the use of semi-supervised learning in selecting articles for systematic reviews. The Experiment 1 results showed that given a small-sized training dataset, semi-supervised methods, especially label spreading, achieved a high level of recall, which makes them viable methods for reducing workload for systematic review article selection. The Experiment 2 results showed that the proposed self-training based method significantly enhanced precision while maintaining a high level of recall. It worked especially well with small training sets (5% or 10% seeds). Next, we explored the feasibility of using active learning to further enhance both recall and precision. Experiment 3 results showed that active learning produced a very high level of recall that meets Cohen et al.'s 95% recall requirement, suggesting that the active learning method is a highly feasible method for systematic review article selection with small-sized training datasets.

4.3. Contributions

In this research, we examined an automated method to classify relevant articles for inclusion or exclusion during the abstract triage stage for creating and updating systematic reviews of medical research. We demonstrated that a novel combination of text analytics techniques, including using the automatically extracted UMLS terms as the features, soft-margin polynomial SVM as the classification algorithm and SMOTE combined with undersampling to deal with the class balance issue, help improve precision while sustaining a high recall (95% or higher) in article classification for SRs. We also demonstrated the viability of semi-supervised learning algorithm along with self-learning and active learning when training dataset is rare, which is often the practical case in many machine learning problems. Our research is intended to make the following contributions.

From a theoretical perspective, this research explores the possibility of combining different text analytics techniques in the area of systematic review development. In prior research, the bag-of-words method has been used as the de facto standard methods for extracting features from article titles and abstract. We used the automatically extracted UMLS terms as features by leveraging the latest version of the MetaMap software and demonstrated that this feature extraction method helps enhance classification performance, as compared with the bag-of-words approach. The class imbalance issue has been insufficiently addressed in extant literature. We explored the use of various re-sampling methods, which have been hardly used in this field, to alleviate the class imbalance problem. We modified SMOTE by combining it with undersampling and used it to enhance article classification performance. This research also explores the feasibility of using a less explored class of machine learning techniques, namely semi-supervised learning, to deal with the small training set problem we often face when creating a new systematic review. In prior research, supervised-learning has been used as the de-facto standard method for article classification for systematic reviews. Supervised learning, however, relies on a large training dataset that in real proactively is extremely costly and time-consuming to obtain. This makes it practically expensive to use supervised learning technique in the case of systematic review creation where a researcher is attempting to answer new medical questions. We proposed to use semi-supervised learning methods such as label spreading, self-training, and active learning to classify articles based on a small-sized training dataset. The use of semi-supervised learning for selecting articles during systematic review creation has so far been largely ignored in literature. The experiences and lessons learned from this research are expected to inform the literature regarding the efficacy of the proposed techniques and the further development and refinement of these techniques

The experiences and lessons learned from our research are expected to inform the literature regarding the efficacy of the proposed techniques and the further development and refinement of these techniques.

From a practical and applied research perspective, this research has the potential to optimize systematic creation and contribute to the adoption of evidence-based medicine. Currently, laborious efforts for selecting articles for systematic reviews preclude us from creating systematic reviews to keep pace with medical research advances, which subsequently impedes the translation of the latest medical evidence into healthcare practice. This research can help to automate the systematic review development process by significantly reducing the number of articles that scientists need to manually review when they create a new systematic review. This research provides direct impact in the availability of best medical evidence and consequently, may contribute to improving the health and wellbeing of society.

4.4. Limitations and future work

Although our research has reached its aims, there are some limitations. First, due of time limitations, our study employed few datasets that have relatively small number of articles (samples) compared to average SR report. Second, this study performed abstract and metadata mining of medical articles, which automate the abstract triage procedure of article selection. However, computerized articles selection techniques could also automate full text-triage procedure of SR generation. Third, we were not able to examine some important machine learning advancement in text mining. Last but not least, is the ability to deploy our proposed machine learning model in a ‘real-life’ setting.

Accordingly, our research can be extended along a number of dimensions. First, the proposed approach can be further evaluated using additional data sets. Probably, datasets derived from the “AHRQ Comparative Effectiveness Reviews”. Second, this approach can be extended to support full-text triage. Nowadays, new big data technologies enable us to deploy algorithms that can easily process not only the abstracts of tens of thousands of articles but also the full text of the articles. Third, future research can use topic-modeling technics like Latent Dirichlet Allocation (LDA) for extraction of abstract features of medical documents, deep-learning techniques where machine learns itself from complex and large-scale dataset. Last but not least, future research may investigate means for deploying the proposed approach in a manner that simplifies and automates (or semi-automate) the update of systematic reviews on a frequent basis as new literature is added to the existing knowledge repository. Other integration and deployment possibilities include the leverage of clinical trials documentation, e.g., from clinicaltrials.gov to further expedite the translation of medical research into practice.

5. References

Adeva, G., Atxa, P., Carrillo, U., & Zengotitabengoa, A. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications, 41*(4), 1498-1508.

Allen, I., & Olkin, I. (1999). Estimating Time to Conduct a Meta - analysis From Number of Citations Retrieved. *JAMA, 282*(7), 634-635.

Ananiadou, S., Procter, R., Rea, B., & Sasaki, Y. (2009). Supporting Systematic Reviews using Text Mining. *3*.

Aronson, A. R., Bodenreider, O., Demner-Fushman, D., Fung, K. W., Lee, V. K., Mork, J. G., et al. From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, 2007* (pp. 105-112): Association for Computational Linguistics

Bekhuis, T., & Demner-Fushman, D. (2012). Screening nonrandomized studies for medical systematic reviews: a comparative study of classifiers. *Artificial intelligence in medicine, 55*, 197-207, doi:10.1016/j.artmed.2012.05.002.

Bennett, K., & Demiriz, A. (1999). Semi-supervised support vector machines. *Advances in Neural Information processing systems*, 368-374.

Chawla, N. V. (2010). DATA MINING FOR IMBALANCED DATASETS: AN OVERVIEW. *Data Mining and Knowledge Discovery Handbook, Springer*.

Cohen, A., Adams, C., Davis, J., Yu, C., Yu, P., Meng, W., et al. (2010). The Essential Role of Systematic Reviews , and the Need for Automated Text Mining Tools. 376-380.

Cohen, A., Ersh, W., & Eterson, K. (2006). Reducing Workload in Systematic Review Preparation Using Automated Citation Classification. 206-219, doi:10.1197/jamia.M1929.The.

Cohen, A. M., Ambert, K., & McDonagh, M. (2009). Cross-topic learning for work prioritization in systematic review creation and update. [Comparative Study
Evaluation Studies
Research Support, N.I.H., Extramural]. *J Am Med Inform Assoc, 16*(5), 690-704, doi:10.1197/jamia.M3162.

Cohen, A. M. C., Ersh, W. R. H., Eterson, K. P., & En, P. O. I. N. Y. (2006a). Reducing Workload in Systematic Review Preparation Using Automated Citation Classification. *JAMIA, 13*, 206-219.

Cohen, A. M. C., Hersh, W. R., Peterson, K., & Yen, P.-Y. (2006b). Reducing Workload in Systematic Review Preparation Using Automated Citation Classification. *JAMIA, 13*(2), 206-219.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Couch, R., Jetha, M., Dryden, D. M., Hooton, N., & Liang, Y. (2008). Diabetes Education for Children With Type 1 Diabetes Mellitus and Their Families. *Agency for Healthcare Research and Quality (US), AHRQ Publication No. 08-E011.*

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation, 10*(7), 1895-1923.

Frunza, O., Inkpen, D., & Matwin, S. (2010). Building Systematic Reviews Using Automatic Text Classification Techniques. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics*, 303-311.

Gieseke, F., Airola, A., Pahikkala, T., & Kramer, O. (2014). Fast and simple gradient-based optimization for semi-supervised support vector machines. *Neurocomputing, 123*, 23-32.

He, H., & M, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*: Technology & Engineering.

Higgins, J., & Green, S. (2011). Cochrane Handbook for Systematic Reviews of Interventions Version 5.1.0 [updated March 2011]. *The Cochrane Collaboration*.

Jin, Y., Huang, C., & Zhao, L. (2011). A Semi-Supervised Learning Algorithm Based on Modified Self-training SVM. *Journal of Computers, 6*(7), 1438-1443.

Joachims, T. (1998a). Text Categorization with Support Vector Machines : Learning with Many Relevant Features. *Universtat Dortmund*, 1-19.

Joachims, T. (1998b). *Text categorization with support vector machines: Learning with many relevant features*: Springer.

Kilicoglu, H., Demner-Fushman, D., Rindflesch, T. C., Wilczynski, N. L., & Haynes, R. B. (2009). Towards automatic recognition of scientifically rigorous clinical research evidence. [Research Support, N.I.H., Intramural]. *J Am Med Inform Assoc, 16*(1), 25-31, doi:10.1197/jamia.M2996.

Kivinen, J., Warmuth, M., & Auer, P. (1995). The perceptron algorithm vs. winnow: Linear vs. logarithmic mistakes bounds when few input variables are relavant. *Conference on Computational Learning Theory*.

Ling, C., & Li, C. (1998). Data Mining for Direct Marketing Problems and Solutions. *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining New York, NY. AAAI Press*.

Liu, A. Y.-c. (2004). *The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets*. The University of Texas at Austin,

Liu, H., Johnson, S. B., & Friedman, C. (2002). Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. [Evaluation Studies

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

Research Support, U.S. Gov't, P.H.S.]. *J Am Med Inform Assoc, 9*(6), 621-636.

Liu, T. Y., Xu, J., Qin, T., Xiong, W., & Li, H. (2007). Letor: Benchmark dataset for research on learning to rank for information retrieval. *In Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, 3-10.

Matwin, S., Kouznetsov, A., Inkpen, D., Frunza, O., & O'Blenis, P. (2010). A new algorithm for reducing the workload of experts in performing systematic reviews. [Research Support, Non-U.S. Gov't]. *J Am Med Inform Assoc, 17*(4), 446-453, doi:10.1136/jamia.2010.004325.

McGowan, J., & Sampson, M. (2005a). Systematic reviews need systematic searchers. *J Med Libr Assoc, 93*(1), 74-80.

McGowan, J., & Sampson, M. (2005b). Systematic reviews need systematic searchers. . *Journal of the Medical Library Association 93*(1), 74-80.

Moher, D., J, T., & AC, T. (2007). Epidemiology and reporting characteristics of systematic reviews. *PLoS Med, 4*(3), e78.

Mulrow, C. (1994). Rationale for systematic reviews. *BMJ, 309*, 597-599.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation, 60*(5), 503-520.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison, 52*(11), 55-66.

Shemilt, I., Simon, A., Hollands, G. J., Marteau, T. M., Ogilvie, D., O'Mara-Eves, A., et al. (2013). Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods*, n/a-n/a, doi:10.1002/jrsm.1093.

Shojania, K. G., Margaret Sampson, Ansari, M. T., & Garritty, C. (2007). Updating Systematic Reviews. *AHRQ, 16*.

Shojania., K. G., Sampson, M., Ansari, M. T., Garritty, C., Doucette, S., Rader, T., et al. (2007). Updating Systematic Reviews. *Agency for Healthcare Research and Quality, Contract No. 290-02-0021*.

Song, M., Yu, H., & Han, W. S. (2011). Combining active learning and semi-supervised learning techniques to extract protein interaction sentences. *BMC bioinformatics, 12*(Suppl 12), S4.

Timsina, P., Liu, J., & El-Gayar, O. (2015). Advanced analytics for the automation of medical systematic reviews. *Information Systems Frontiers*(A Special Issue on Big Data and Analytics in Healthcare), 1-16.

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., & Coiera, E. (2014). Systematic review automation technologies. *Syst Rev, 3*, 74, doi:10.1186/2046-4053-3-74.

US National Library of Medicine (2014). Unified Medical Language System® (UMLS®). http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html2014.

Wallace, B. C., Trikalinos, T. a., Lau, J., Brodley, C., & Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC bioinformatics, 11*, 55, doi:10.1186/1471-2105-11-55.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Sch¨olkopf, B. (2004). Learning with Local and Global Consistency. *Max Planck Institute for Biological Cybernetics, 72076 Tuebingen, Germany*.

Zhu, X. (2005). Semi-supervised learning literature survey. TR-1530. *University of Wisconsin-Madison Department of Computer Science*.

Zhu, X., & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. *Technical Report CMU-CALD-02-107, Carnegie Mellon University*.

Appendix

6. Publication Resulted From This Dissertation

6.1. Journals

Timsina, P., Liu, J, & El-Gayar, O. (2015). Text Analytics for Automation of Medical Systematic Review Creation and Update, Information System Frontiers

Liu, J Timsina, P., & El-Gayar (nd). Semi-supervised Article Selection for Medical Systematic Reviews: the Case of a Small-sized Training Dataset, Information System Frontiers (Submitted)

6.2. Conferences

Timsina, P., & Liu, J El-Gayar, Shan, Y. (2015). Using Semi-supervised Learning for the Creation of Medical Systematic Review: An Exploratory Analysis. Americas Conference of Information System, IEEE: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, HICSS-49 2016

Timsina, P., El-Gayar., & Liu, J (2015). Active Learning For Automation of Knowledge Generation: The Case of Rare and Expensive Training Dataset. Americas Conference of Information System, Puerto Rico, Aug 2015

Timsina, P., El-Gayar., & Liu, J (2015). Leveraging Advanced Analytics Techniques for Medical Systematic Review Update. IEEE: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, HICSS-48 2015

Timsina, P., El-Gayar, O., & Nawar, N. (2014). Leveraging Advanced Analytics to Generate Dynamic Medical Systematic Reviews. *Twentieth Americas Conference on Information Systems,*

Savannah, 2014.

El-Gayar, O., & Timsina, P. (2014). *Opportunities for Business Intelligence and Big Data Analytics In Evidence Based Medicine*. Paper presented at the IEEE: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, HICSS-47 2014 Conference January 6-9, 2014.

Timsina, P., El-Gayar, O., & Nawar, N. (2014). Information Technology for Evidence Based Medicine: Status and Future Direction. *Twentieth Americas Conference on Information Systems, Savannah, 2014*.